A021 838

REPORT

NCSL 242-75

OCTOBER 1975

THE ANALYSIS OF A LONGITUDINAL CONTROL SYSTEM FOR UNDERWATER VEHICLES

D. E. HUMPHREYS

R. W. MILLER

L. F. DEWBERRY

NAVAL COASTAL SYSTEMS LABORATORY

NCSL

PANAMA CITY, FLORIDA 32401

copy- 81

NCSL

NAVAL COASTAL SYSTEMS LABORATORY

PANAMA CITY, FLORIDA 32401

ADMINISTRATIVE INFORMATION

This effort was accomplished jointly under Task Area Number ZF 61-112-001, Work Unit Number 00112-50-1A, Vehicle Hydrodynamics and Task Area Number SF 34-371-208, Work Unit Number 16436, Analytical Investigation of Countermeasure Launchers.

APPROVED OCTOBER 1975
W. T. ODUM, Head
Diving and Salvage Department

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NCSL 242-75	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE ANALYSIS OF LONGITUDINAL CONTROL SYSTEM FOR UNDERWATER VEHICLES		5. TYPE OF REPORT & PERIOD COVERED Informal
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Douglas E. Humphreys Richard W. Miller Larry F. Dewberry		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Coastal Systems Laboratory Panama City, Florida 32401		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Tasks SF 34-371-208 and ZF 61-112-001 Work Units 00112-50-1A&16436
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE October 1975
		13. NUMBER OF PAGES 72
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Underwater vehicles	Automatic control	Models
Longitudinal stability	Towed vehicles	
Feedback control	Hydrodynamic coefficients	
Computer programs	Hydromechanics	
Self-propelled vehicles	Frequency	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A general longitudinal feedback control system containing pitch, pitch rate, depth, and depth rate feedback is described. The Laplace domain transfer functions for each feedback loop are developed. A computer analysis program utilizing the root locus technique is developed for aiding in the design of the control system. An illustrative example design problem is included.

19.

Identifiers

Multiloop control system
Longitudinal feedback system
Vehicle longitudinal feedback

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

INTRODUCTION

The requirement for feedback control systems in underwater vehicles is well established. In its absolute form, feedback systems are used to control vehicle depth and yaw, while rate feedback (pitch or yaw rate) can be used to improve a vehicle's handling characteristics. The purpose of this report is to present a step-by-step procedure for the analysis of vehicle longitudinal feedback control systems.

The control system discussed is general in that it allows the designer to select any or all of four feedback loops (pitch rate, pitch, depth rate, and depth). Each loop is analyzed separately for this purpose. The root-locus technique is used in the analysis. To aid the engineer in the design process, a computer program has been written that will perform all the necessary computations. This program is suitable for analyzing both self-propelled and towed vehicles. Inputs to the program consists of vehicle length, speed, mass, moments of inertia, and the 30 linear hydrodynamic coefficients. Vehicle mass and moments of inertia are computed using the MIDCOHV computer program WTBAL reported in NCSL Report 220-74[1]. The hydrodynamic coefficients are computed in the MIDCOHV computer program GEORGE. The details necessary for running the program are presented in the users guide section. An example design is included to illustrate the analysis of a longitudinal feedback control system. The analysis of lateral feedback control systems is discussed in an NCSL report[2].

---

[1] Naval Coastal Systems Laboratory Report 220-74, *The MIDCOHV Weight and Balance Computer Program (WTBAL)*, by K. W. Watkinson, September 1974, Unclassified.

[2] Naval Coastal Systems Laboratory Report, *The Analysis of Lateral Control Systems for Self-Propelled and Towed Submarged Vehicles*, by Douglas E. Humphreys, Richard W. Miller, and Larry F. Dewberry, (in publication), Unclassified

LONGITUDINAL CONTROL SYSTEM ANALYSIS

BACKGROUND

A general model of a longitudinal feedback system is shown in Figure 1. The feedback loops are pitch rate ($\dot{\theta}$), pitch angle (θ), depth rate ($\dot{Z}$), and depth (Z). The sensors are modeled as pure gains and are denoted as $K_{\dot{\theta}}$, K_{θ}, $K_{\dot{Z}}$, and K_Z. There are two command inputs: desired depth (Z_o) and desired pitch angle (θ_o).

The purpose of a feedback control system is to either stabilize an unstable system, improve the system response characteristics, to control a certain variable, such as depth, or a combination of these. The desired vehicle control is achieved by successively closing each loop and varying the loop gain until the desired system dynamics are acheived. The root-locus method is used here to aid in the analysis process. For additional details on the root-locus method and the mathematics of Laplace transforms, See References 3, 4, 5, and 6 and Appendix A.

FIRST LOOP

The inner-most loop (or first loop) is shown in Figure 2. The vehicle transfer function relating pitch rate response to stern plane input is

$$\frac{\dot{\theta}}{\delta_s} = \frac{s\theta}{\delta_s} = \frac{N^{\theta}_{\delta_s} s}{D} .$$

---

[3]Clark, R. N., *Introduction to Automatic Control Systems,* John Wiley and Sons, Inc., 1973.

[4]Hale, F. J., *Introduction to Control System Analysis and Design,* Prentice-Hall, Inc., 1973.

[5]Blakelock, J. H., *Automatic Control of Aircraft and Missiles*, John Wiley and Sons, Inc., 1965.

[6]Hildebrand, F. B., *Advanced Calculus for Applications*, Prentice-Hall, Inc., 1963.

(Text Continued on Page 4)

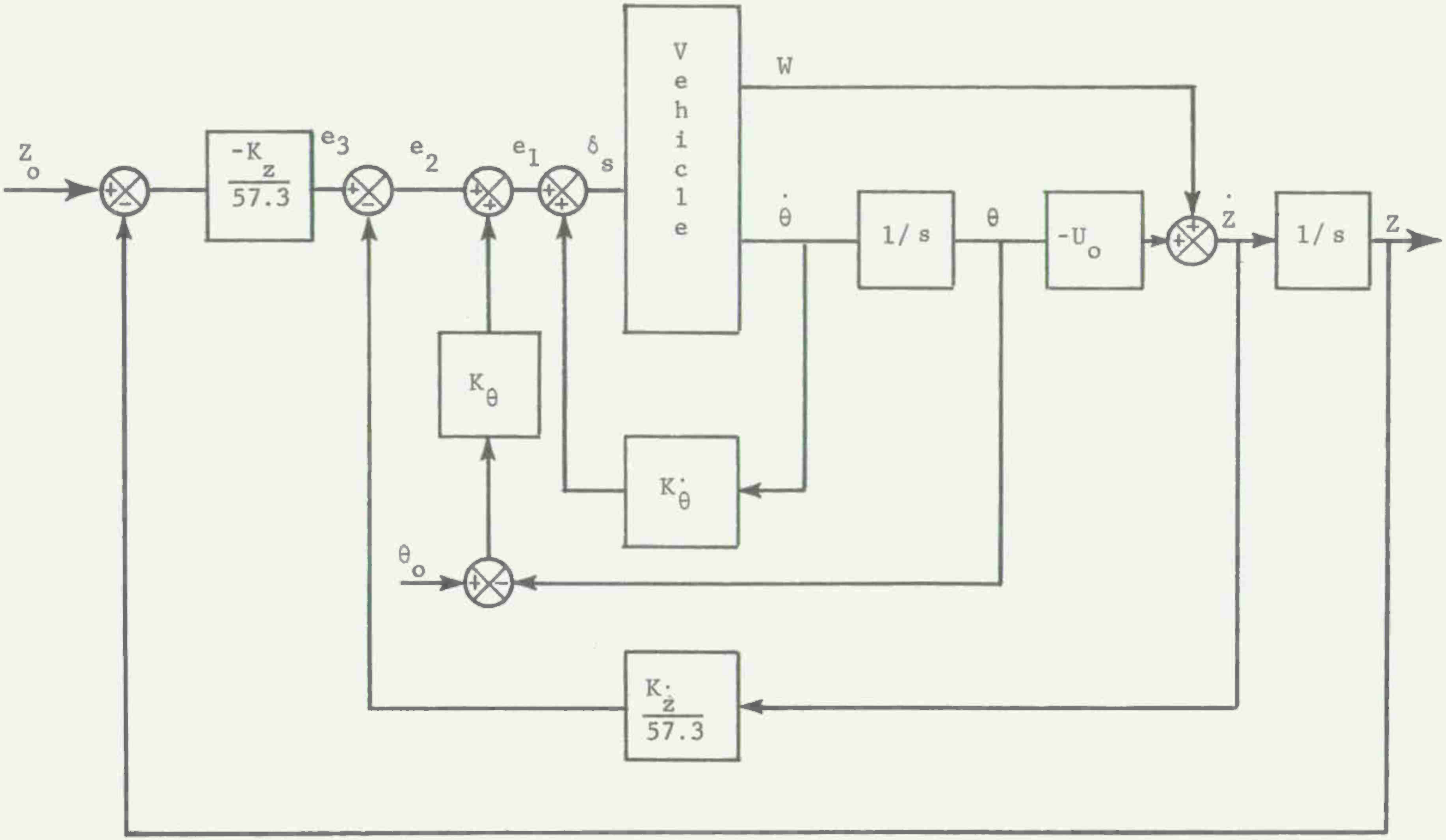


FIGURE 1. BLOCK DIAGRAM FOR LONGITUDINAL CONTROL SYSTEM

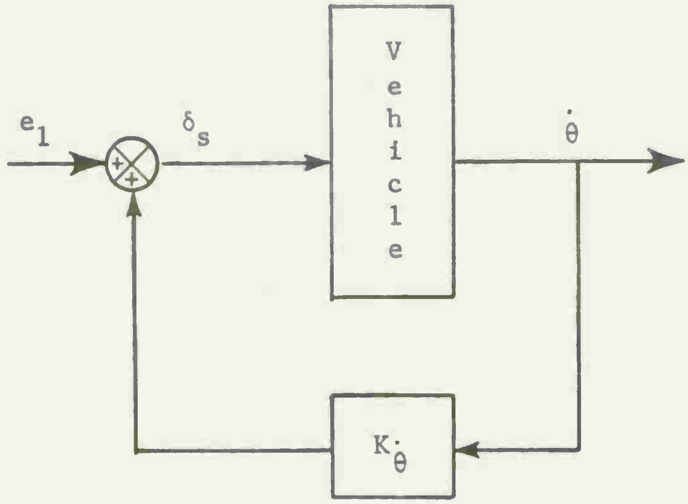


FIGURE 2. BLOCK DIAGRAM FOR THE FIRST LOOP

Where $N_{\delta_s}^{\theta}$ = Pitch angle/control deflection transfer function numerator

D = Denominator of vehicle transfer function.

The numerator and denominator are functions of the vehicle hydrodynamic coefficients. Appendix B gives the expanded form of each of the vehicle transfer functions.

By solving for the closed loop transfer function, $\dot{\theta}/e_1$, and varying the feedback gain, $K_{\dot{\theta}}$, the system dynamics can be adjusted to yield the desired performance. Solving for this closed loop transfer function yields

$$\frac{\dot{\theta}}{e_1} = \frac{\dfrac{sN_{\delta_s}^{\theta}}{D}}{1 - \dfrac{sN_{\delta_s}^{\theta}}{D} K_{\dot{\theta}}} = \frac{sN_{\delta_s}^{\theta}}{D - sN_{\delta_s}^{\theta} K_{\dot{\theta}}} \; . \quad \checkmark$$

For stability: $K_{\dot{\theta}} \geq 0$.

Note that although Figure 2 shows the feedback signal being added to the input signal, the system is actually a negative feedback system since the numerator, $N_{\delta_s}^{\theta}$ *, will always carry a negative sign.*

Closing the first loop yields a new vehicle; i.e., a rate controlled vehicle, with a new characteristic equation

$$D' = D - sN^{\theta}_{\delta_s} K_{\dot{\theta}} .$$

The single prime indicates a system with one loop closure.

SECOND LOOP

Figure 3 shows the vehicle second loop after the first loop has been closed. The vehicle transfer function with one loop closed is denoted by $\dot{\theta}/e_1$. The vehicle transfer functions with two loops closed is

$$\frac{\theta}{e_2} = \frac{1/s \ \frac{\dot{\theta}}{e_1}}{1 + \frac{\dot{\theta}}{e_1} \frac{K_\theta}{s}} = \frac{N^{\theta}_{\delta_s}}{D - sN^{\theta}_{\delta_s} K_{\dot{\theta}} + K_\theta N^{\theta}_{\delta_s}}$$

For stability: $K_\theta \leq 0$.

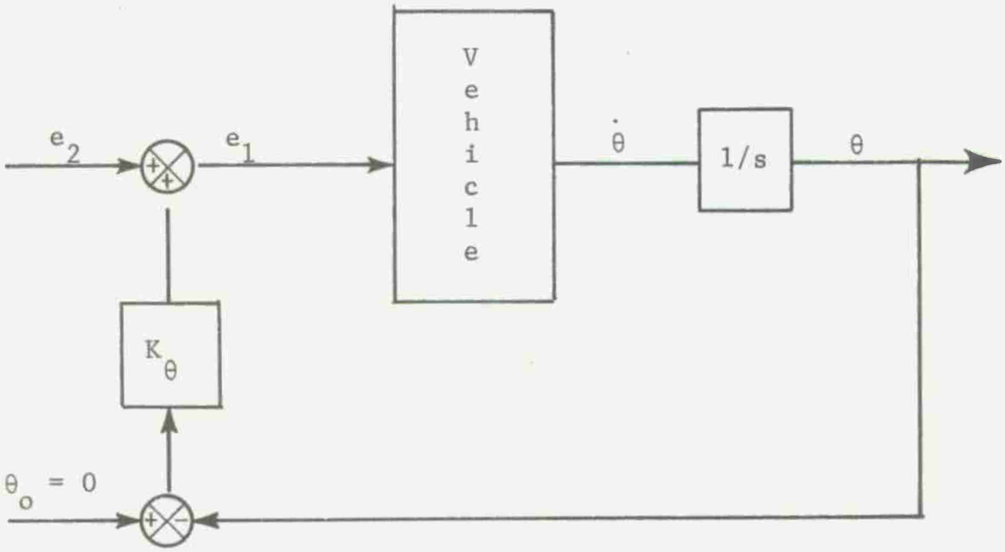


FIGURE 3. BLOCK DIAGRAM FOR THE SECOND LOOP

Note that in order to solve for the above transfer function, the pitch angle command was set to zero. Since the commanded pitch angle does not affect the vehicle characteristic response, this requirement in no way restricts the analysis capability. After the system response has been evaluated for $\theta_o = 0$, trajectories for other pitch angle commands can be evaluated using time domain solutions such as the one shown in Reference 7.

THIRD LOOP

Figure 4 shows the block diagram of the third loop. The vehicle transfer function with two loops closed is denoted by θ/e_2. To form the depth rate signal ($\dot{Z}$) requires the combination of the pitch signal and the vertical velocity according to the following equation

$$\dot{Z} = w - U_o\theta \ ,$$

or

$$\dot{Z}/\theta = w/\theta - U_o .$$

The w/θ transfer function is obtained by dividing the w/δ_s transfer function by the θ/δ_s transfer function which yields

$$\frac{w/\delta_s}{\theta/\delta_s} = \frac{w}{\theta} = \frac{\dfrac{N^w_{\delta_s}}{D}}{\dfrac{N^\theta_{\delta_s}}{D}} = \frac{N^w_{\delta_s}}{N^\theta_{\delta_s}} \ .$$

The theoretical basis for this operation can be found in an NSRDC report[7].

---

[7]Naval Ship Research and Development Center, Report No. P-4-3-H-01, *User's Guide NSRDC Digital Program for Simulating Submarine Motion ZZMN Revision 1.0,* by Ronald W. Richard, June 1971, Unclassified.

The vehicle transfer function with three loops closed is

$$\frac{\dot{Z}}{e_3} = \frac{\frac{\theta}{e_2}\,(\frac{w}{\theta} - U_o}{1 + \frac{\theta}{e_2}\,(\frac{w}{\theta} - U_o)\,K_{\dot{Z}}}$$

$$\frac{\dot{Z}}{e_3} = \frac{N^w_{\delta_s} - N^\theta_{\delta_s} U_o}{D - sN^\theta_{\delta_s} K_{\dot{\theta}} + K_\theta N_\theta + K_{\dot{Z}}\,(N^w_{\delta_s} - N^\theta_{\delta_s} U_o)}\ .$$

For stability $K_{\dot{Z}} \geq 0$.

Note that from Figure 4 this is a *positive feedback system*. This convention was chosen to conform with the Navy's standard motion simulation program in Hildebrand's textbook[6]. The reader should note the difference between a negative feedback and a positive feedback root locus. In a *negative* feedback system, the locus of roots on the real axis lies to the *left* of an odd number of poles or zeros. In a *positive* feedback system, the locus of roots on the real axis lies to the *right* of an odd number of poles or zeros. In *both* cases, the locus emanates from a pole and terminates at a zero.

Also note from Figure 4 that the value for $K_{\dot{Z}}$ is dimensionalized by dividing it by 57.3 (≃ 4 arctan 1).

FOURTH LOOP

Figure 5 shows the model of the fourth and final loop. The vehicle transfer function with three loops closed is denoted by $\dot{Z}/e_3$. The vehicle transfer function with four loops closed is

$$\frac{Z}{Z_o} = \frac{-K_Z \frac{\dot{Z}}{e_3} \frac{1}{s}}{1 - K_Z \frac{\dot{Z}}{e_3} \frac{1}{s}}$$

$$\frac{Z}{Z_o} = \frac{-K_Z(N^w_{\delta_s} - N_{s}\, U_o)}{S[D - sN^\theta_{\delta_s} K_{\dot{\theta}} + K_\theta N^\theta_{\delta_s} + K_{\dot{Z}}\,(N^w_{\delta_s} - N^\theta_{\delta_s} U_o)] - K_Z\,(N^w_{\delta_s} - N^\theta_{\delta_s} U_o)}\ .$$

---

(6) ibid.

(Text Continued on Page 9)

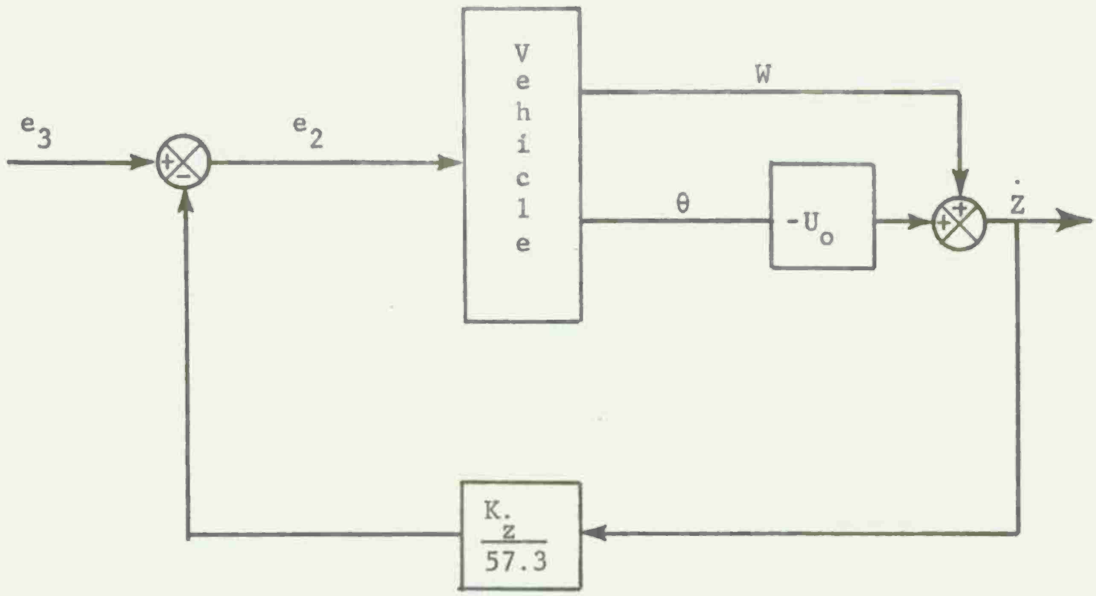


FIGURE 4. BLOCK DIAGRAM FOR THE THIRD LOOP

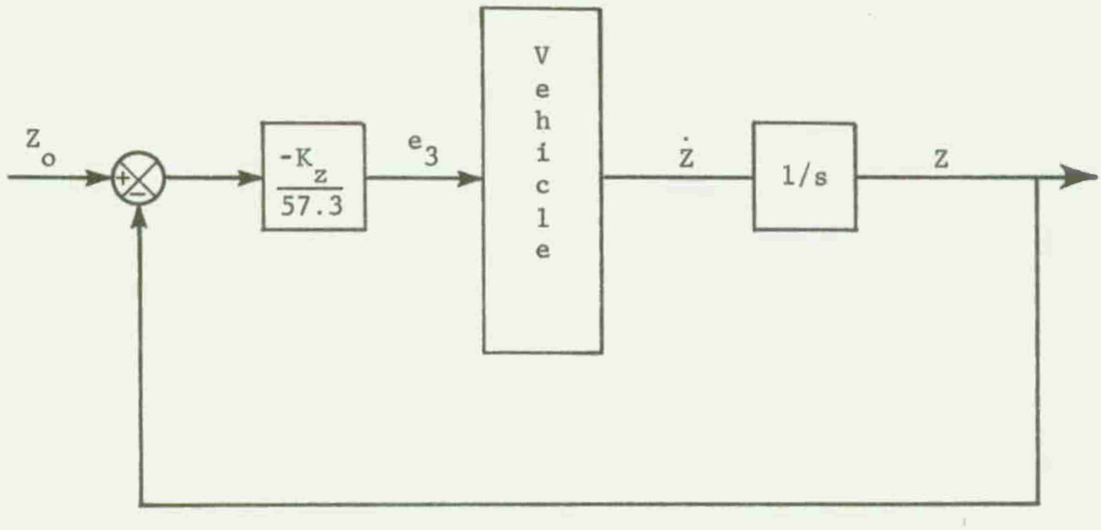


FIGURE 5. BLOCK DIAGRAM FOR THE FOURTH LOOP

For stability: $K_Z \leq 0$.

Note again that this is *positive feedback*. Again notice that the value of K_Z is dimensionalized by dividing it by 57.3.

The augmented vehicle response is achieved by the commanded deflection of the stern plane. The control law that determines the stern plane position as a function of time is seen from Figure 1 as

$$\delta_s = K_{\dot{\theta}} \dot{\theta} = (\theta_o - \theta) K_\theta - K_{\dot{z}} \dot{Z} - (Z_o - Z) K_Z .$$

This conforms to the control law used in Reference 8.

COMPUTER PROGRAM USERS GUIDE

BASIC PROGRAM DESCRIPTION

This program computes the roots of the numerator and denominator for a submerged vehicle with the feedback control system described in the previous section and Appendix C. The basic inputs to the program are the vehicle nondimensional hydrodynamic coefficients, mass, and moments of inertia as defined in a SNAME publication[9]. The vehicle length and speed, whether the vehicle is a towed or self-propelled body (inputed by means of a disk data file), and the range of loop gains to be analyzed (inputed systematically from an interactive terminal).

The main program computes the coefficients of the numerator and denominator equations that are shown in Appendix B. This calculation is broken down into four basic segments; one segment for each loop analyzed. The resulting polynomial equations are solved for the roots of the system by using a polynomial root factoring routine.

At the completion of the main analysis program, a data file can be automatically written for a companion program known as TIMEPLT (Appendix D). TIMEPLT is another analysis program which takes the S-domain

---

(8) Air Force Flight Dynamics Laboratory Research and Technology Division, Wright-Patterson Air Force Base, *Analysis of Multiloop Vehicular Control Systems,* by D. T. McRuer, I. L. Ashkenas, and H. R. Pass, March 1964.

(9) Society of Naval Architects and Mechanical Engineers, *Nomenclature for Treating the Motion of a Submerged Body through a Fluid,* 1952.

analysis and transfers it back into the time domain, complete with plot diagrams of the system output response due to standard input signals.

INPUT REQUIREMENTS

Data input to the object program LØCSAP/ØBJECT is through the disk data file LØCSAP/DATA and from an interactive terminal. All inputs are in a free-field format. Refer to Figure 6 for an example of data file input.

INTERACTIVE TERMINAL DATA

As described earlier, the control system analysis proceeds by closing each successive feedback loop. This is accomplished by inputting a range of loop gains for the innermost loop and then deciding on a single value before proceeding to the next system loop. At any point in the analysis, the operator is allowed to change any previous loop gains until a full set of four loop gains have been chosen. Details of the interactive terminal data can be found in the example problem (Figure 7).

EXAMPLE PROBLEM*

The program is executed as follows:

```
??EX LØCSAP/ØBJECT * [charge number]

    FILE FILE 1 = LØCSAP/DATA;END.
```

At the interactive terminal, the programer is now allowed to have outputed three different data sets. The program prints the statement:

```
PRINT STAB DERIV, DIMRTS, TFCPRT . . .
```

These data sets are the vehicle nondimensional stability derivatives (input to the program from the previously mentioned data file LØCSAP/DATA), the dimensional polynomial roots, and the dimensional polynomial coefficients, respectively. A "1" input for each of the variables allows the data to be printed out; an "0" means that the printout is not desired. Following these data sets, the numerator roots (zeros) for the four control system loops are printed.

---

*This example problem is for a self-propelled vehicle, consequently some additional terms are zeroed by the program.

(Text Continued on Page 14)

Line Number	Data Description		
Record 1	0000:10.1270, 49.3330, 0,*		
Record 2	0010: -.15020E-01,	-.57700E-03,	.81000E-04,/
	0020: .0 ,	-.50138E-01,	.95500E-02,/
	0030: .0 ,	.0 ,	-.15709E 00,/
	0040: .0 ,	-.17455E-01,	-.11310E-01,/
	0050: -.16230E-02,	.0 ,	.0 ,/
	0060: .0 ,	-.31545E-01,	-.14600E-03,/
	0070: .0 ,	-.13000E-03,	-.15730E-02,/
	0080: .0 ,	.0 ,	.0 ,/
	0090: .0 ,	.0 ,	.0 ,/
	0100: .0 ,	-.27695E-01,	-.12797E-01,/
	0110: .36397E-01,	.19170E-02,	

Note: The following information identifies the input by record, column, and line.

Record 1	0000:	UQ, LB, [Type] UØ - Vehicle Velocity (ft/sec) LB - Vehicle length (ft) [Type] - 1 if a towed body, 0 if self-propelled		
Record 2*	0010:	XU ,	ZU ,	MU
	0020:	XW ,	ZW ,	MW
	0030:	XTHUSQ**,	ZTHUSQ**,	MTHUSQ**
	0040:	XQ ,	ZQ ,	MQ
	0050:	XUD ,	ZUD ,	MUD
	0060:	XWD ,	ZWD ,	MWD
	0070:	XQD ,	ZQD ,	MQD
	0080:	XX ,	ZX ,	MX
	0090:	XZ ,	ZZ ,	MZ
	0100:	XDELT ,	ZDELT ,	MDELT
	0110:	M, IY ,		

*Nondimensional Stability Derivatives (Appendix A)

**Program reads XTHUSQ($=X'_\theta U_o^2$), ZTHUSQ, MTHUSQ and converts to XTH(= XTHUSQ/U_o^2), ZTH, MTH

FIGURE 6. EXAMPLE DATA FILE

```
  5:LOCSAP/ØBJECT = 1 BOJ 1302 10/01/74 FROM 01/06
PRINT STAB DERIV. DIMRTS, TFCPRT...
1,1,1←
```

S N A M E NON-DIMENSIONAL
LONGITUDINAL STABILITY DERIVATIVES

XU = -.15020E-01	ZU = -.57700E-03	MU = .81000E-04
XW = .0	ZW = -.50138E-01	MW = .95500E-02
XTH = .0	ZTH = .0	MTH = .15317E-02
XQ = .0	ZQ = -.17455E-01	MQ = -.11310E-01
XUD = -.16230E-02	ZUD = .0	MUD = .0
XWD = .0	ZWD = -.31545E-01	MWD = -.14600E-03
XQD = .0	ZQD = -.13000E-03	MQD = -.15730E-02
XX = .0	ZX = .0	MX = .0
XZ = .0	ZZ = .0	MZ = .0
XDELT = .0	ZDELT = -.27695E-01	MDELT = -.12797E-01

M .36397E-01 IY = .19170E-02

**** DENOMINATOR DS(J) ****

DIMENSIONAL COEFFICIENTS

J = 1	DS = .0
J = 2	DS = .0
J = 3	DS = .273739387358E-04
J = 4	DS = .118877286995E-02
J = 5	DS = .185102157604E-01
J = 6	DS = .108587460950E 00
J = 7	DS = .120470690538E 00

DIMENSIONAL ROOTS

J = 1	ROOTR = -.59231E-01	ROOTI = -.22004E-01
J = 2	ROOTR = -.59231E-01	ROOTI = .22004E-01
J = 3	ROOTR = -.81096E-01	ROOTI = .0
J = 4	ROOTR = -.70180E 00	ROOTI = .0
J = 5	ROOTR = .0	ROOTI = .0
J = 6	ROOTR = .0	ROOTI = .0

**** X NUMERATOR ****
***** XS(J) COEFFICIENTS ALL ZERO *****

FIGURE 7. EXAMPLE PROBLEM
(Sheet 1 of 2)

**** Z NUMERATOR ****

DIMENSIONAL COEFFICIENTS

J =	1	ZS =	.0
J =	2	ZS =	-.153127243398E-03
J =	3	ZS =	-.116585064408E-01
J =	4	ZS =	-.128614635109E 00
J =	5	ZS =	-.100337496561E 00

DIMENSIONAL ROOTS

J =	1	ROOTR =	-.15883E-01	ROOTI =	.0
J =	2	ROOTR =	-.81096E-01	ROOTI =	.0
J =	3	ROOTR =	-.11848E 01	ROOTI =	.0
J =	4	ROOTR =	.0	ROOTI =	.0

**** I NUMERATOR ****

DIMENSIONAL COEFFICIENTS

J =	1	TS =	.0
J =	2	TS =	.0
J =	3	TS =	-.322969154535E-03
J =	4	TS =	-.548520303593E-02
J =	5	TS =	-.185293864643E-01

DIMENSIONAL ROOTS

J =	1	ROOTR =	-.81096E-01	ROOTI =	.0
J =	2	ROOTR =	-.21493E 00	ROOTI =	.0
J =	3	ROOTR =	.0	ROOTI =	.0
J =	4	ROOTR =	.0	ROOTI =	.0

ZEROS OF TD/E1
-0.0811 0.0000 -0.2149 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000

ZEROS OF T/E2
-0.0811 0.0000 -0.2149 0.0000 0.0000 0.0000 0.0000 0.0000

ZEROS OF ZD/E3 AND Z/ZO
-0.0811 0.0000 -0.3690 0.0000 1.0384 0.0000 0.0000 0.0000
0.0000 0.0000

FIGURE 7.
(Sheet 2 of 2)

FIRST LOOP ANALYSIS

To analyze the first loop (pitch rate), the initial value of $K_{\dot{\theta}}$ must be entered along with the step increment, $\Delta K_{\dot{\theta}}$, and the final value of $K_{\dot{\theta}}$. The program prints out the statement:

ENTER KTDØRG, DELKTD, KTDFIN.

The programmer must then enter the desired values as follows:

0, .5, 5.

A root locus for the inner loop will then be generated for the gain values of 0 to 5 in steps of .5 (Figure 8).

```
ENTER KTDORG,DELKTD,KTDFIN

0,.5,5←
   0.00
 -0.0592  -0.0220  -0.0592   0.0220  -0.0811   0.0000  -0.7018   0.0000
  0.0000   0.0000   0.0000   0.0000
   0.50
 -0.0407   0.0000  -0.0811   0.0000  -0.0897   0.0000  -0.7667   0.0000
  0.0000   0.0000   0.0000   0.0000
   1.00
 -0.0307   0.0000  -0.0811   0.0000  -0.1094   0.0000  -0.8340   0.0000
  0.0000   0.0000   0.0000   0.0000
   1.50
 -0.0253   0.0000  -0.0811   0.0000  -0.1227   0.0000  -0.9030   0.0000
  0.0000   0.0000   0.0000   0.0000
   2.00
 -0.0217   0.0000  -0.0811   0.0000  -0.1328   0.0000  -0.9734   0.0000
  0.0000   0.0000   0.0000   0.0000
   2.50
 -0.0190   0.0000  -0.0811   0.0000  -0.1409   0.0000  -1.0449   0.0000
  0.0000   0.0000   0.0000   0.0000
   3.00
 -0.0170   0.0000  -0.0811   0.0000  -0.1475   0.0000  -1.1172   0.0000
  0.0000   0.0000   0.0000   0.0000
   3.50
 -0.0154   0.0000  -0.0811   0.0000  -0.1531   0.0000  -1.1901   0.0000
  0.0000   0.0000   0.0000   0.0000
   4.00
 -0.0141   0.0000  -0.0811   0.0000  -0.1578   0.0000  -1.2637   0.0000
  0.0000   0.0000   0.0000   0.0000
   4.50
 -0.0129   0.0000  -0.0811   0.0000  -0.1618   0.0000  -1.3376   0.0000
  0.0000   0.0000   0.0000   0.0000
   5.00
 -0.0120   0.0000  -0.0811   0.0000  -0.1654   0.0000  -1.4119   0.0000
  0.0000   0.0000   0.0000   0.0000
```

FIGURE 8. EXAMPLE PROBLEM: FIRST LOOP ANALYSIS

Each root is listed as a real and imaginary pair, read from left to right, top and bottom. For example, at a gain value $K_{\dot{\theta}} = 0.00$ (first three digit number printed) the denominator roots (poles) are

Real	Imaginary
-.0592	-j .0220
-.0592	+j .0220
-.0811	j0
-.7018	j0

While the loop zeros (previously printed out) are

Real	Imaginary
-.0811	j0
-.2149	j0
0.0	j0.0

The two zero vlaue roots for each are not shown since this example is for a self-propelled vehicle; for a towed vehicle they would have a nonzero value.

Figure 9 is a plot of the root locus for this loop.

After printing the value for the roots over the gain range specified, the program will then print out the next statement:

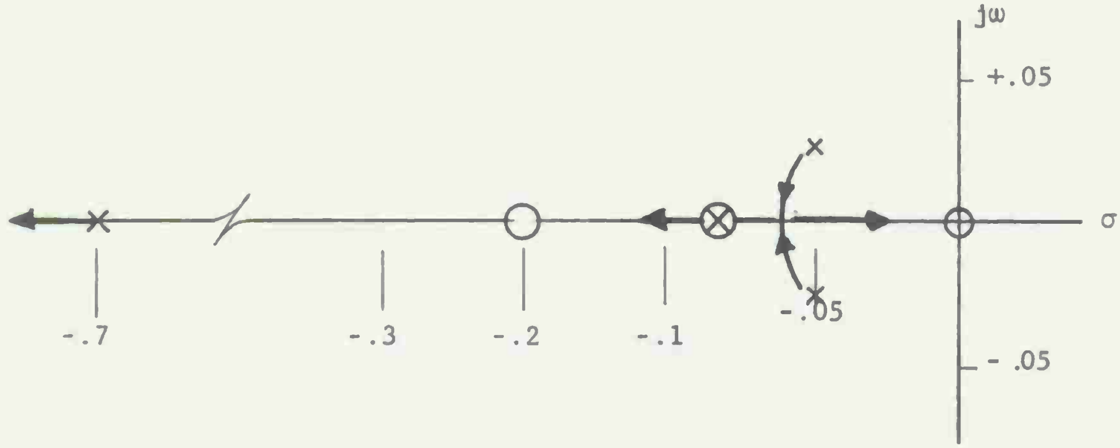


FIGURE 9. ROOT LOCUS FOR THE FIRST LOOP

DØ YØU WANT TØ CØTINUE KTD R.LØCUS.

The programmer must enter:

0 - if he desires to go on to the second loop.

1 - if he desires to continue the root locus in the first loop.

If 1 is entered, the program will again ask for values of KTDØRG, DELKTD, KTDFIN.

If 0 is entered, the program will ask for the desired first loop gain by printing ENTER KTD. The programmer must enter the selected first loop gain.

SECOND, THIRD, AND FOURTH LOOPS

The above procedure is repeated for the next three loops. When new values are requested at the end of each root locus, three options are available. This is to allow the programmer to go back to any desired inside loop at any point in the program. For example, if after completing the root locus in the second loop, the programmer may wish to reanalyze the first loop based on what he learned from the second loop root locus. This is accomplished as follows. The program prints out the statement

DO / YØU WANT NEW KTD, 0 = NØ, 1 = ENT, 2 = ENT & CØMP, KTD NØW = 1

after the second loop root locus is complete. The programmer must enter

0 - If he desires to go to the third loop,

1 - If he desires to change the value of the first loop gain,

2 - If he desires to go back to the first loop and recompute the root locus.

This return option is available at the completion of each loop's locus.

For the fourth loop, two gain values are printed, the loop gain and the system gain. The system gain is defined as

$$\text{GAIN} = K_Z \frac{ZS(5)}{DS(5)}$$

for self-propelled vehicles (see Appendix B for definition of terms). The gain is the steady state value for depth, Z, for a unit step in depth command, Z_o. Normally, it is desirable for this gain value to be equal to unity.

The remainder of the vehicle control system analysis follows as shown in an example problem (Figure 10), along with root locus plots for the second, third, and fourth loops (Figures 11, 12, and 13).

When the programmer is satisfied with the analysis and has selected gain values for all four loops, the program prints out

DØ YØU WANT A TIMEPLT.

By inputing a 1 (yes) to the LØCSAP program, a data file is automatically created which will later be utilized by the program TIMEPLT. The root values to be passed to the file are determined by the operators answer to the next two questions asked of him by the program.

ENTER KTD, KT, KZD, KZ, GAMP

TIMEPLT FØR WHICH LØØP .

The K values are the individual loop gains, and the variable GAMP is indicative of the type and magnitude of the standard test signal to be inputed to the system. Answering the question as to which loop, determines which loop output will be plotted by the companion program. This analysis can be repeated as many times as desired, until the operator answers NØ (0) to the question:

DØ YØU WANT ANØTHER TIMEPLT,

at which point the root-locus program is terminated.

CONCLUSIONS

A computer program was written utilizing the root-locus technique to analyze a longitudinal feedback control system. An example problem is included illustrating the use of this program.

To date, this analysis program has been utilized in the design of control systems for swimmer delivery vehicles, a submarine, and towed mine-hunting vehicles.

```
DO YOU WANT TO CONTINUE KTD R. LOCUS
0←
ENTER KTD
1←
ENTER KTORG,DELKT,KTFIN
0,-.5,-5←
   0.00
 -0.0307   0.0000  -0.0811   0.0000  -0.1094   0.0000  -0.8340   0.0000
  0.0000   0.0000   0.0000   0.0000
  -0.50
 -0.1150  -0.1130  -0.1150   0.1130  -0.0811   0.0000  -0.7441   0.0000
  0.0000   0.0000   0.0000   0.0000
  -1.00
 -0.1724  -0.1651  -0.1724   0.1651  -0.0811   0.0000  -0.6292   0.0000
  0.0000   0.0000   0.0000   0.0000
  -1.50
 -0.0811   0.0000  -0.2584  -0.2186  -0.2584   0.2186  -0.4572   0.0000
  0.0000   0.0000   0.0000   0.0000
  -2.00
 -0.0811   0.0000  -0.3110   0.0000  -0.3315  -0.3342  -0.3315   0.3342
  0.0000   0.0000   0.0000   0.0000
  -2.50
 -0.0811   0.0000  -0.2726   0.0000  -0.3508  -0.4364  -0.3508   0.4364
  0.0000   0.0000   0.0000   0.0000
  -3.00
 -0.0811   0.0000  -0.2565   0.0000  -0.3588  -0.5185  -0.3588   0.5185
  0.0000   0.0000   0.0000   0.0000
  -3.50
 -0.0811   0.0000  -0.2476   0.0000  -0.3632  -0.5888  -0.3632   0.5888
  0.0000   0.0000   0.0000   0.0000
 -4.00
 0.0811    0.0000  -0.2410   0.0000   0.3661   0.6514  -0.3661   0.6514
 0.0000    0.0000   0.0000   0.0000
  -4.50
 -0.0811   0.0000  -0.2379   0.0000  -0.3681   -0.7083  -0.3681   0.7083
  0.0000   0.0000   0.0000   0.0000
  -5.00
 -0.0811   0.0000  -0.2349   0.0000  -0.3696  -0.7609  -0.3696   0.7609
  0.0000   0.0000   0.0000   0.0000
DO YOU WANT NEW KTD,0=NO,1=ENT,2=ENT&COMP,KTD NOW=     1.00
0←
DO YOU WANT TO CONTINUE KT R. LOCUS
0←
```

FIGURE 10. EXAMPLE PROBLEM: SECOND, THIRD AND FOURTH LOOP ANALYSIS
(Sheet 1 of 5)

```
ENTER KT
-1←
ENTER KZDORG,DELKZD,KZDFIN
0,.5,5←
    0.00
  -0.1724  -0.1651  -0.1724   0.1651  -0.0811   0.0000  -0.6292   0.0000
   0.0000   0.0000   0.0000   0.0000
    0.50
  -0.0811   0.0000  -0.3992   0.0000  -0.0792  -0.6952  -0.0792   0.6952
   0.0000   0.0000   0.0000   0.0000
    1.00
  -0.0811   0.0000  -0.3839   0.0000   0.1213  -0.9539   0.1213   0.9539
   0.0000   0.0000   0.0000   0.0000
    1.50
  -0.0811   0.0000  -0.3788   0.0000   0.3270  -1.1186   0.3270   1.1186
   0.0000   0.0000   0.0000   0.0000
    2.00
  -0.0811   0.0000  -0.3764   0.0000   0.5340  -1.2271   0.5340   1.2271
   0.0000   0.0000   0.0000   0.0000
    2.50
  -0.0811   0.0000  -0.3749   0.0000   0.7415  -1.2938   0.7415   1.2938
   0.0000   0.0000   0.0000   0.0000
    4.00
  -0.0811   0.0000  -0.3739   0.0000   0.9492  -1.3249   0.9492   1.3249
   0.0000   0.0000   0 0000   0.0000
    3.50
  -0.0811   0.0000  -0.3732   0.0000   1.1571  -1.3229   1.1571   1.3229
   0.0000   0.0000   0.0000   0.0000
    3.00
  -0.0811   0.0000  -0.3726   0.0000   1.3650  -1.2877   1.3650   1.2877
   0.0000   0.0000   0.0000   0.0000
    4.50
  -0.0811   0.0000  -0.3722   0.0000   1.5731  -1.2164   1.5731   1.2164
   0.0000   0.0000   0.0000   0.0000
    5.00
  -0.0811   0.0000  -0.3719   0.0000   1.7811  -1.1019   1.7811   1.1019
   0.0000   0.0000   0.0000   0.0000
DO YOU WANT NEW KTD, KTD NOW =    1.00
0←
DO YOU WANT NEW KT, KT NOW =   -1.00
0←
DO YOU WANT TO CONTINUE KZD R. LOCUS
1←
ENTER KZDORG,DELKZD,KZDFIN
0,-.5,-3←
```

FIGURE 10.
(Sheet 2 of 5)

```
  0.00
 -0.1724  -0.1651  -0.1724   0.1651  -0.0811   0.0000  -0.6292   0.0000
  0.0000   0.0000   0.0000   0.0000
 -0.50
 -0.0811   0.0000   0.2740   0.0000  -0.3410   0.0000  -1.3235   0.0000
  0.0000   0.0000   0.0000   0.0000
 -1.00
 -0.0811   0.0000  -0.3547   0.0000   0.4253   0.0000  -1.8776   0.0000
  0.0000   0.0000   0.0000   0.0000
 -1.50
 -0.3594   0.0000   0.5174   0.0000  -0.0811   0.0000  -2.3814   0.0000
  0.0000   0.0000   0.0000   0.0000
 -2.00
 -0.0811   0.0000  -0.3617   0.0000   0.5822   0.0000  -2.8603   0.0000
  0.0000   0 0000   0.0000   0.0000
 -2.50
 -0.0811   0.0000  -0.3632   0.0000   0.6311   0.0000  -3.3242   0.0000
  0.0000   0.0000   0.0000   0.0000
 -3.00
 -0.0811   0.0000  -0.3641   0.0000   0.6698   0.0000  -3.7783   0.0000
  0.0000   0.0000   0.0000   0.0000
DO YOU WANT NEW KTD, KTD NOW =     1.00
0←
DO YOU WANT NEW KT, KT NOW =    -1.00
0←
DO YOU WANT TO CONTINUE KZD R. LOCUS
0←
ENTER KZD
0,.5,5←
   0.00
GAIN =   0.000
 -0.1724  -0.1651  -0.1724   0.1651  -0.0811   0.0000  -0.6292   0.0000
  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
   0.50
GAIN =  -2.710
 -0.0811   0.0000  -0.3803   0.0000   0.4495   0.0000  -0.5216  -0.8133
 -0.5216   0.8133   0.0000   0.0000   0.0000   0.0000
   1.00
GAIN =  -5.421
 -0.0811   0.0000  -0.3745   0.0000   0.5577   0.0000  -0.5787  -1.0922
 -0.5787   1.0922   0.0000   0.0000   0.0000   0.0000
   1.50
GAIN =  -8.131
 -0.0811   0.0000   0.6230   0.0000  -0.3726   0.0000  -0.6122  -1.2989
 -0.6122   1.2989   0.0000   0.0000   0.0000   0.0000
```

FIGURE 10.
(Sheet 3 of 5)

```
   2.00
GAIN =  -10.841
  -0.0811   0.0000  -0.3717   0.0000   0.6690   0.0000  -0.6357  -1.4705
  -0.6357   1.4705   0.0000   0.0000   0.0000   0.0000
   2.50
GAIN = -13.552
  -0.0811   0.0000  -0.3711   0.0000   0.7041   0.0000  -0.6535  -1.6204
  -0.6535   1.6204   0.0000   0.0000   0.0000   0.0000
   3.00
 GAIN = -16.262
  -0.0811   0.0000  -0.3708   0.0000   0.7321   0.0000  -0.6677  -1.7552
  -0.6677   1.7552   0.0000   0.0000   0.0000   0.0000
   3.50
GAIN = -18.972
  -0.0811   0.0000  -0.3705   0.0000   0.7552   0.0000  -0.6794  -1.8788
  -0.6794   1.8788   0.0000   0.0000   0.0000   0.0000
   4.00
GAIN = -21.683
  -0.0811   0.0000  -0.3703   0.0000   0.7746   0.0000  -0.6892  -1.9937
  -0.6892   1.9937   0.0000   0.0000   0.0000   0.0000
   4.50
GAIN = -24.393
 -0.0811   0.0000  -0.3702   0.0000   0.7913   0.0000  -0.6976  -2.1014
 -0.6976   2.1014   0.0000   0.0000   0.0000   0.0000
   5.00
GAIN = -27.103
  -0.0811   0.0000  -0.3701   0.0000   0.8058   0.0000  -0.7049  -2.2032
  -0.7049   2.2032   0.0000   0.0000   0.0000   0.0000
DO YOU WANT NEW KTD, KTD NOW =      1.00
0←
DO YOU WANT NEW KT, KT NOW =    -1.00
0←
DO YOU WANT NEW KZD,KZD NOW =        0.00
0←
DO YOU WANT TO CONTINUE KZ R. LOCUS
1←
ENTER KZORG,DELKZ,KZFIN
0,-.5,-3←
   0.00
GAIN =   0.000
   -0.1724  -0.1651  -0.1724   0.1651  -0.0811   0.0000  -0.6292   0.0000
    0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
   -0.50
GAIN =    2.710
   -0.0811   0.0000  -0.3586   0.0000   0.3000  -0.5254   0.3000   0.5254
   -1.2154   0.0000   0.0000   0.0000   0.0000   0.0000
   -1.00
```

FIGURE 10.
(Sheet 4 of 5)

```
GAIN =    5.421
   -0.0811   0.0000  -0.3637   0.0000   0.4504  -0.6146   0.4504   0.6146
   -1.5113   0.0000   0.0000   0.0000   0.0000   0.0000
    -1.50
GAIN =    8.131
   -0.0811   0.0000   0.5629  -0.6620   0.5629   0.6620  -0.3654   0.0000
   -1.7345   0.0000   0.0000   0.0000   0.0000   0.0000
    -2.00
GAIN =   10.841
   -0.0811   0.0000   0.6564  -0.6901   0.6564   0.6901  -0.3663   0.0000
   -1.9206   0.0000   0.0000   0.0000   0.0000   0.0000
    -2.50
GAIN =   13.552
    0.7380  -0.7066   0.7380   0.7066  -0.0811   0.0000  -0.3668   0.0000
   -2.0832   0.0000   0.0000   0.0000   0.0000   0.0000
    -3.00
GAIN =   16.262
   -0.0811   0.0000  -0.3672   0.0000   0.8112  -0.7151   0.8112   0.7151
   -2.2293   0.0000   0.0000   0.0000   0.0000   0.0000

DO YOU WANT NEW KTD,KTD NOW =     1.00
0←
DO YOU WANT NEW KT, KT NOW =   -1.00
0←
DO YOU WANT NEW KZD,KZD NOW  =    0.00
0←
DO YOU WANT TO CONTINUE KZ R. LOCUS
0←
DO YOU WANT A TIMEPLT
1←
ENTER KTD,KT,KZD,KZ,GAMP
1,-1,0,0,10←
TIMEPLT FOR WHICH LOOP; ENTER 1=1ST, 2=2ND, 3=3RD, 4=4TH
2←
DO YOU WANT ANOTHER TIMEPLT
0←

PROCESSOR TIME   =     15 SEC      $  0.60
I/O TIME         =     23 SEC      $  0.46
PRORATED TIME    =    249 SEC      $  2.49
                      TOTAL COST   $  3.55

LOCSAP/MILLER = 1 EOJ  1326
```

FIGURE 10.
(Sheet 5 of 5)

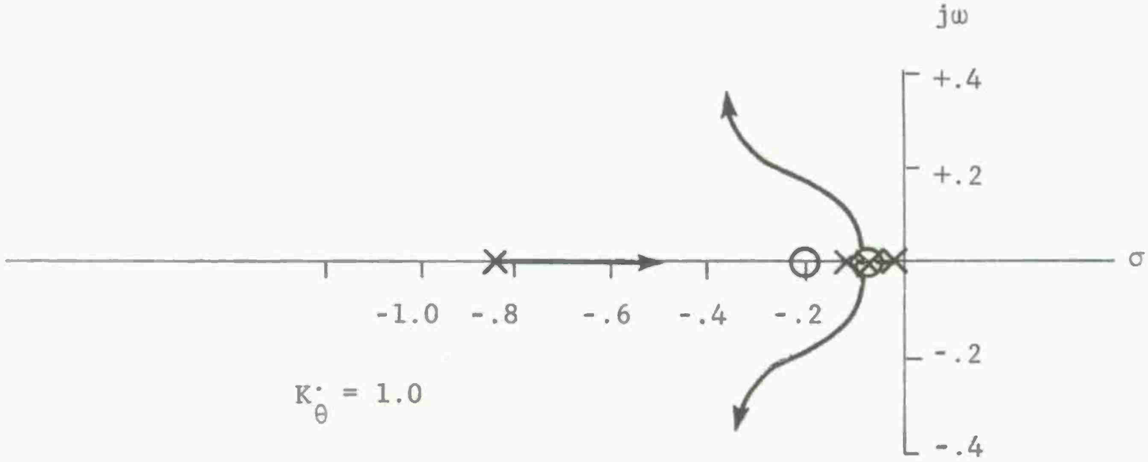


FIGURE 11 ROOT LOCUS FOR THE SECOND LOOP

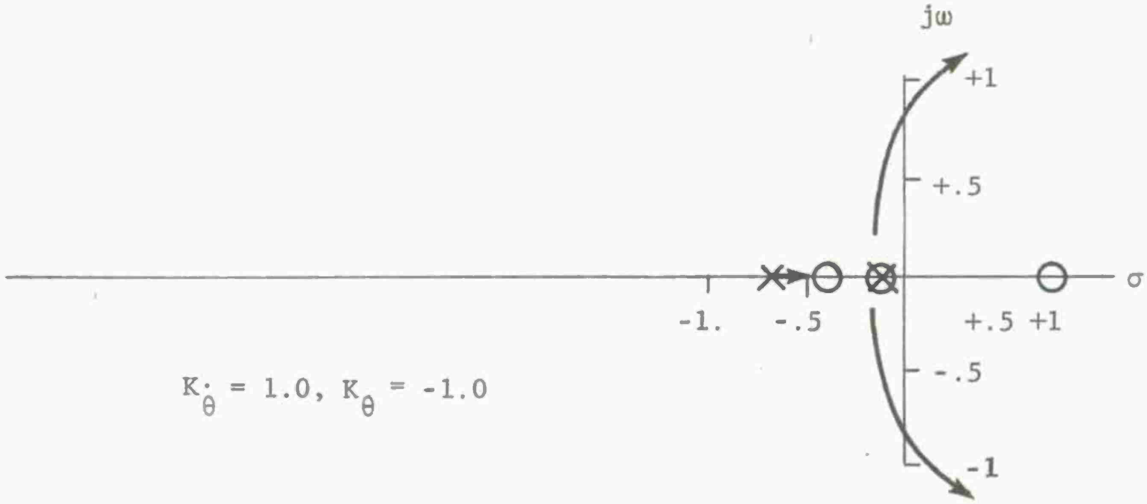


FIGURE 12. ROOT LOCUS FOR THE THIRD LOOP

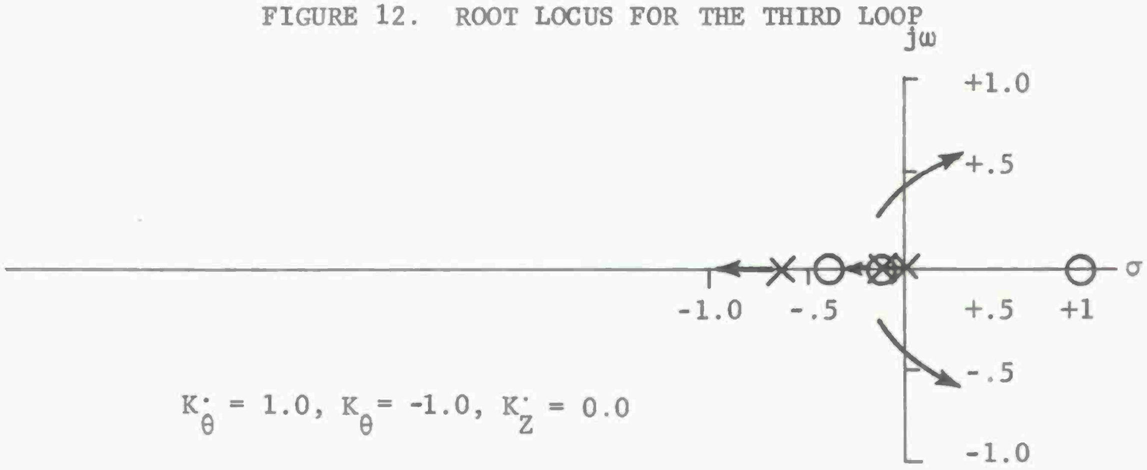


FIGURE 13. ROOT LOCUS FOR THE FOURTH LOOP

APPENDIX A

SOME NOTES ON THE CONSTRUCTION AND INTERPRETATION OF ROOT LOCUS

A physical system can be represented by a block diagram composed of individual blocks that represent the various components of the system as shown in Figure A1. Each block is described by one or more

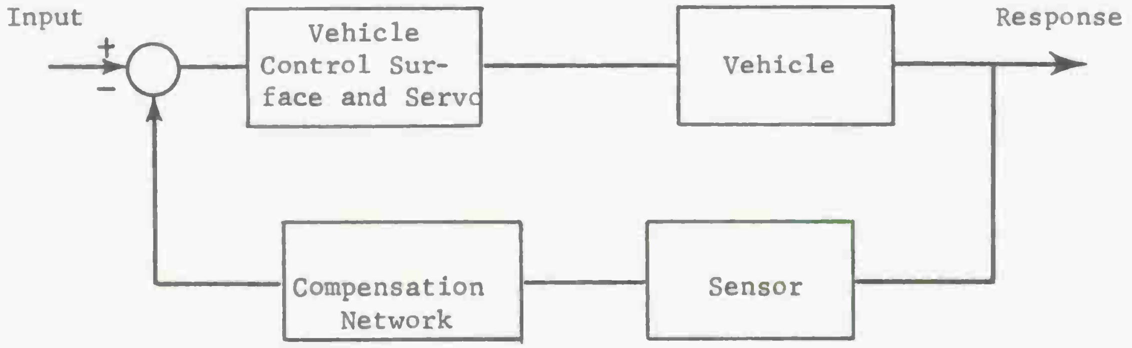


FIGURE A1. A TYPICAL BLOCK DIAGRAM

differential equations according to Newton's Second Law of Motion or its electrical equivalent. Combining the characteristics of each block to form the characteristics of the overall system is quite difficult because a signal is modified in both phase and amplitude in going through each block. By applying the Laplace transform

$$F(s) = \int_0^\infty f(t)e^{-st}\,dt$$

to the describing differential equations, one obtains an algebraic representation for each block in the system. It is then convenient to arrange this representation in the form of a transfer function; i.e., as a ratio of block response to block excitation. These transfer functions can be be multiplied together to yield system response to system excitation. It is a relatively straightforward procedure because each transfer function is merely a ratio of polynomials in the Laplace operator s.

Once the transfer functions for each of the blocks have been formed, it is then necessary to examine the effects of changing various unknown system parameters, such as feedback gain on the system dynamics. The root-locus diagram was developed to facilitate such an analysis. As the name implies, it shows on one figure the trajectory that the frequency and damping characteristic modes follow as system parameters are changed.

Consider the transfer function

$$\frac{O}{I} = \frac{K(s + a)(s^2 + bs + c)}{s(s + d)(s + e)(s^2 + fs + g)} .$$

The denominator of the transfer function represents the characteristic equation of the system; e.g., the equation describing the free motion of the system (the response independent of control input). It is responsible for the general solution of the system of differential equations. The particular solution comes from the numerator.

It will be observed that all values of s which make the denominator zero are solutions of the characteristic equation and therefore contirbute a term of the $e^{\lambda t}$ to the time response. Since for these roots the transfer function is undefined, denominator roots are called poles. Numerator roots are appropriately called zeros. It is customary to plot these poles and zeros on a graph whose abscissa is the real part of s and whose ordinate is the imaginary part. Poles are commonly depicted as x's and zeros are O's. A first order root; e.g., $(s + d)$, will always lie on the abscissa. A second order system has two roots. They may be real, in which case they lie on the abscissa, or they may be complex, in which case they are placed equidistant above and below the abscissa.

Any pole which lies in the right half s-plane represents an unstable motion. Zeros in the right half plane are significant in terms of the type motion only if the system depicted is a feedback system. In this case the zeros represent the location of the poles when the feedback gain is made infinite. For zeros in the right half plane then, the system will become unstable at some finite value of feedback gain. Knowledge of the location of the basic vehicle zeros is needed by designers to combine the control system characteristics with those of the vehicle to obtain the desired response without unexpected instabilities. Note also that a zero placed on top of a pole will eliminate the motion caused by that pole from the time history of the particular variable associated with the numerator (θ in θ/δ_r) for example) but from no other time history.

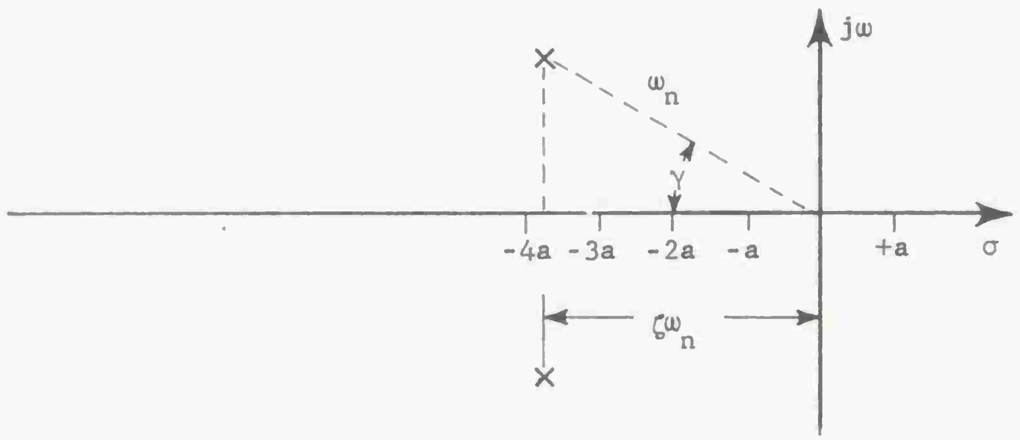
A pole located at $s = -3$, for example, means that there is contribution to the time history given by e^{-3t}. Thus, the further to the left the pole, the more rapid is the subsidence. Conversely, a pole at $s = 3$ means the motion has an unstable component described by e^{3t}.

Stable oscillatory modes, it will be recalled, have roots which can be expressed by

$$s_1, s_2 = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$$

Figure A2 indicates how varying either frequency or damping ratio separately moves the poles. It also shows that the product $\zeta\omega_n$ determines the time for an oscillation to decay to half amplitude. When $\zeta\omega_n = 0.591$, the oscillation will decay to half amplitude in 1 second. Smaller values of the product mean the time to damp to half amplitude is longer.

For further details on the construction and interpretation of root locus diagrams refer to *Introduction to Automatic Control Systems*, John Wiley and Son, 1962.



ζ = Damping Ratio = $\cos \gamma$

ω_n = Undamped Natural Frequency

$\zeta\omega_n$ = Total Damping

t_{ss} = Time to reach 0.95 steady state value = $3.0/\zeta\omega_n$

$t_{1/2}$ = Time to damp to half amplitude = $0.69/\zeta\omega_n$

t_2 = Time to double amplitude (unstable systems only) = $0.69/\zeta\omega_n$

FIGURE A2. IMPORTANT FEATURES OF ROOT-LOCUS DIAGRAM

(Reverse Page A-4 Blank)

APPENDIX B

EXPRESSIONS FOR THE LONGITUDINAL TRANSFER FUNCTION COEFFICIENTS

The longitudinal characteristic equation is

$$\Delta_{\text{Long}} = As'^4 + Bs'^3 + Cs'^2 + Ds' + E$$

where

$$\begin{aligned}
A = {} & (m' - X'_{\dot u})(m' - Z'_{\dot w})(I'_y - M'_{\dot q}) - X'_{\dot w} M'_{\dot u} Z'_{\dot q} \\
& - Z'_{\dot u} M'_{\dot w} X'_{\dot q} - (m' - Z'_{\dot w}) M'_{\dot u} X'_{\dot q} \\
& - Z'_{\dot u} X'_{\dot w} (I'_y - M'_{\dot q}) - M'_{\dot w} (m' - X'_{\dot u}) Z'_{\dot q} \; .
\end{aligned}$$

$$\begin{aligned}
B = {} & -(m' - X'_{\dot u})(m' - Z'_{\dot w}) M'_q - Z'_w (m' - X'_{\dot u})(I'_y - M'_{\dot q}) \\
& - X'_u (m' - Z'_{\dot w})(I'_y - M'_{\dot q}) - X'_{\dot w} M'_{\dot u} (Z'_q + m') \\
& - M'_u X'_{\dot w} Z'_{\dot q} - M'_{\dot u} X'_w Z'_{\dot q} - Z'_{\dot u} M'_{\dot w} X'_q - Z'_{\dot u} M'_w X'_{\dot q} \\
& - Z'_u M'_{\dot w} X'_{\dot q} - (m' - Z'_{\dot w}) M'_{\dot u} X'_q - M'_u (m' - Z'_{\dot w}) X'_{\dot q} \\
& + M'_{\dot u} Z'_w X'_{\dot q} + Z'_{\dot u} X'_{\dot w} M'_q - Z'_u X'_{\dot w} (I'_y - M'_{\dot q}) \\
& - Z'_{\dot u} X'_w (I'_y - M'_{\dot q}) - M'_{\dot w} (m' - X'_{\dot u})(Z'_q + m') \\
& - M'_w (m' - X'_{\dot u}) Z'_{\dot q} + M'_{\dot w} X'_u Z'_{\dot q} \; .
\end{aligned}$$

$$
\begin{aligned}
C = \; & -(m' - X'_{\dot{u}})(m' - Z'_{\dot{w}})M'_{\theta} + Z'_{w}(m' - X'_{\dot{u}})M'_{q} \\
& + X'_{u}(m' - Z'_{\dot{w}})M'_{q} + Z'_{w}X'_{u}(I'_{y} - M'_{\dot{q}}) - X'_{\dot{w}}M'_{\dot{u}}Z'_{\theta} \\
& - M'_{u}X'_{\dot{w}}(Z'_{q} + m') - M'_{\dot{u}}X'_{w}(Z'_{q} + m') - Z'_{\dot{u}}M'_{\dot{w}}X'_{\theta} \\
& - Z'_{\dot{u}}M'_{w}X'_{q} - Z'_{u}M'_{\dot{w}}X'_{q} - Z'_{u}M'_{w}X'_{\dot{q}} \\
& - (m' - Z'_{\dot{w}})M'_{\dot{u}}X'_{\theta} - M'_{u}(m' - Z'_{\dot{w}})X'_{q} + M'_{\dot{u}}Z'_{w}X'_{q} \\
& + Z'_{w}M'_{u}X'_{\dot{q}} + Z'_{\dot{u}}X'_{\dot{w}}M'_{\theta} + Z'_{u}X'_{\dot{w}}M'_{q} + Z'_{\dot{u}}X'_{w}M'_{q} \\
& - X'_{w}Z'_{u}(I'_{y} - M'_{\dot{q}}) - M'_{\dot{w}}(m' - X'_{\dot{u}})Z'_{\theta} - M'_{w}(m' - X'_{\dot{u}})(Z'_{q} + m') \\
& + M'_{\dot{w}}X'_{u}(Z'_{q} + m') + X'_{u}M'_{w}Z'_{\dot{q}} - M'_{u}X'_{w}Z'_{\dot{q}} \; .
\end{aligned}
$$

$$
\begin{aligned}
D = \; & Z'_w M'_\theta (m' - X'_{\dot{u}}) + X'_u (m' - Z'_{\dot{w}}) M'_\theta - Z'_w X'_u M'_q \\
& - M'_u X'_{\dot{w}} Z'_\theta - M'_{\dot{u}} X'_w Z'_\theta - M'_u X'_w (Z'_q + m') \\
& - Z'_{\dot{u}} M'_w X'_\theta - Z'_u M'_{\dot{w}} X'_\theta - Z'_u M'_w X'_q \\
& - M'_u (m' - Z'_{\dot{w}}) X'_\theta + M'_{\dot{u}} Z'_w X'_\theta + Z'_w M'_u X'_q \\
& + Z'_u X'_{\dot{w}} M'_\theta + Z'_{\dot{u}} X'_w M'_\theta + X'_w Z'_u M'_q \\
& - M'_w (m' - X'_{\dot{u}}) Z'_\theta + M'_{\dot{w}} X'_u Z'_\theta + M'_w X'_u (Z'_q + m') \; .
\end{aligned}
$$

$$
\begin{aligned}
E = \; & - Z'_w X'_u M'_\theta - M'_u X'_w Z'_\theta - Z'_u M'_w X'_\theta \\
& + Z'_w M'_u X'_\theta + X'_w Z'_u M'_\theta + X'_u M'_w Z'_\theta \; .
\end{aligned}
$$

The pitch response transfer function is

$$\frac{\theta}{\delta_s} = \frac{N^{\theta}_{\delta_s}}{\Delta_{Long}} = \frac{A_\theta s'^2 + B_\theta s' + C_\theta}{\Delta_{Long}}$$

where

$$A_\theta = M'_{\delta_e}(m' - X'_{\dot{u}})(m' - Z'_{\dot{w}}) + Z'_{\delta_e} X'_{\dot{w}} M'_{\dot{u}} + X'_{\delta_e} Z'_{\dot{u}} M'_{\dot{w}}$$
$$- M'_{\delta_e} X'_{\dot{w}} Z'_{\dot{u}} + X'_{\delta_e}(m' - Z'_{\dot{w}}) M'_{\dot{u}} + Z'_{\delta_e} M'_{\dot{w}}(m' - X'_{\dot{u}}) .$$

$$B_\theta = -M'_{\delta_e}(m' - X'_{\dot{u}}) Z'_{w} - M'_{\delta_e} X'_{u}(m' - Z'_{\dot{w}}) + Z'_{\delta_e} X'_{\dot{w}} M'_{u}$$
$$+ Z'_{\delta_e} X'_{w} M'_{\dot{u}} + X'_{\delta_e} Z'_{\dot{u}} M'_{w} + X'_{\delta_e} Z'_{u} M'_{\dot{w}}$$
$$- M'_{\delta_e} X'_{\dot{w}} Z'_{u} - M'_{\delta_e} X'_{w} Z'_{\dot{u}} + X'_{\delta_e}(m' - Z'_{\dot{w}}) M'_{u}$$
$$- X'_{\delta_e} Z'_{w} M'_{\dot{u}} - Z'_{\delta_e} M'_{\dot{w}} X'_{u} + Z'_{\delta_e} M'_{w}(m' - X'_{\dot{u}}) .$$

$$C_\theta = M'_{\delta_e} X'_{u} Z'_{w} + Z'_{\delta_e} X'_{w} M'_{u} + X'_{\delta_e} Z'_{u} M'_{w}$$
$$- M'_{\delta_e} X'_{w} Z'_{u} - X'_{\delta_e} Z'_{w} M'_{u} - Z'_{\delta_e} M'_{w} X'_{u} .$$

The vertical velocity transfer function is

$$\frac{W'}{\delta_s} = \frac{N^W_{\delta_s}}{\Delta_{Long}} = \frac{A_W s'^3 + B_W s'^2 + C_W s' + D_W}{\Delta_{Long}}$$

where

$$A_w = Z'_{\delta_e}(m' - X'_{\dot{u}})(I'_y - M'_{\dot{q}}) + X'_{\delta_e} M'_{\dot{u}} Z'_{\dot{q}} + M'_{\delta_e} Z'_{\dot{u}} X'_{\dot{q}}$$

$$- Z'_{\delta_e} M'_{\dot{u}} X'_{\dot{q}} + X'_{\delta_e} Z'_{\dot{u}}(I'_y - M'_{\dot{q}}) + M'_{\delta_e}(m' - X'_{\dot{u}}) Z'_{\dot{q}} \quad .$$

$$B_w = - Z'_{\delta_e}(m' - X'_{\dot{u}}) M'_q - Z'_{\delta_e} X'_u (I_y - M'_{\dot{q}})$$

$$+ X'_{\delta_e} M'_u Z'_{\dot{q}} + X'_{\delta_e} M'_{\dot{u}} (Z'_q + m')$$

$$+ M'_{\delta_e} Z'_{\dot{u}} X'_q + M'_{\delta_e} Z'_u X'_{\dot{q}}$$

$$- Z'_{\delta_e} M'_{\dot{u}} X'_q - Z'_{\delta_e} M'_u X'_{\dot{q}}$$

$$- X'_{\delta_e} Z'_{\dot{u}} M'_q + X'_{\delta_e} Z'_u (I'_y - M'_{\dot{q}})$$

$$+ M'_{\delta_e}(m' - X'_{\dot{u}})(Z'_q + m') - M'_{\delta_e} X'_u Z'_{\dot{q}} \quad .$$

$$
\begin{aligned}
C_w = & -Z'_{\delta_e}(m' - X'_{\dot{u}})M'_{\theta} + Z'_{\delta_e}X'_u M'_q + X'_{\delta_e}M'_{\dot{u}}Z'_{\theta} \\
& + X'_{\delta_e}M'_u(Z'_q + m') + M'_{\delta_e}Z'_{\dot{u}}X'_{\theta} + M'_{\delta_e}Z'_u X'_q \\
& - Z'_{\delta_e}M'_{\dot{u}}X'_{\theta} - Z'_{\delta_e}M'_u X'_q - X'_{\delta_e}Z'_{\dot{u}}M'_{\theta} - X'_{\delta_e}Z'_u M'_q \\
& + M'_{\delta_e}(m' - X'_{\dot{u}})Z'_{\theta} - M'_{\delta_e}X'_u(Z'_q + m') .
\end{aligned}
$$

$$
\begin{aligned}
D_w = & \; Z'_{\delta_e}X'_u M'_{\theta} + X'_{\delta_e}M'_u Z'_{\theta} + M'_{\delta_e}Z'_u X'_{\theta} \\
& - Z'_{\delta_e}M'_u X'_{\theta} - X'_{\delta_e}Z'_u M'_{\theta} - M'_{\delta_e}X'_u Z'_{\theta} .
\end{aligned}
$$

The forward speed transfer function is

$$\frac{U'}{\delta_s} = \frac{N^U_{\delta_s}}{\Delta_{Long}} = \frac{A_U s'^3 + B_U s'^2 + C_U s' + D_U}{\Delta_{Long}}$$

where

$$A_u = X'_{\delta_e}(m' - Z'_{\dot{w}})(I'_y - M'_{\dot{q}}) + M'_{\delta_e} X'_{\dot{w}} Z'_{\dot{q}} + Z'_{\delta_e} M'_{\dot{w}} X'_{\dot{q}}$$
$$+ M'_{\delta_e}(m' - Z'_{\dot{w}}) X'_{\dot{q}} + Z'_{\delta_e} X'_{\dot{w}} (I'_y - M'_{\dot{q}}) - X'_{\delta_e} M'_{\dot{w}} Z'_{\dot{q}} \quad .$$

$$B_u = -X'_{\delta_e}(m' - Z'_{\dot{w}}) M'_q - X'_{\delta_e}(I'_y - M'_{\dot{q}}) Z'_w$$
$$+ M'_{\delta_e} X'_{\dot{w}} (Z'_q + m') + M'_{\delta_e} X'_w Z'_{\dot{q}} + Z'_{\delta_e} M'_{\dot{w}} X'_q$$
$$+ Z'_{\delta_e} M'_w X'_{\dot{q}} + M'_{\delta_e}(m' - Z'_{\dot{w}}) X'_q - M'_{\delta_e} Z'_w X'_{\dot{q}}$$
$$- Z'_{\delta_e} X'_{\dot{w}} M'_q + Z'_{\delta_e} X'_w (I'_y - M'_{\dot{q}})$$
$$- X'_{\delta_e} M'_{\dot{w}} (Z'_q + m') - X'_{\delta_e} M'_w Z'_{\dot{q}} \quad .$$

$$C_{u} = -X'_{\delta_e}(m'-Z'_{\dot{w}})M'_{\theta} + X'_{\delta_e}Z'_{w}M'_{q} + M'_{\delta_e}X'_{\dot{w}}Z'_{\theta}$$

$$+ M'_{\delta_e}X'_{w}(Z'_{q}+m') + Z'_{\delta_e}M'_{\dot{w}}X'_{\theta} + Z'_{\delta_e}M'_{w}X'_{q}$$

$$+ M'_{\delta_e}(m'-Z'_{\dot{w}})X'_{\theta} - M'_{\delta_e}Z'_{w}X'_{q} - Z'_{\delta_e}X'_{\dot{w}}M'_{\theta}$$

$$- Z'_{\delta_e}X'_{w}M'_{q} - X'_{\delta_e}M'_{\dot{w}}Z'_{\theta} - X'_{\delta_e}M'_{w}(Z'_{q}+m') \; .$$

$$D_{u} = X'_{\delta_e}Z'_{w}M'_{\theta} + M'_{\delta_e}X'_{w}Z'_{\theta} + Z'_{\delta_e}M'_{w}X'_{\theta}$$

$$- M'_{\delta_e}Z'_{w}X'_{\theta} - Z'_{\delta_e}X'_{w}M'_{\theta} - X'_{\delta_e}M'_{w}Z'_{\theta} \; .$$

APPENDIX C

LONGITUDINAL CONTROL SYSTEM ANALYSIS PROGRAM - CARD LIST

(Reverse Page C-2 Blank)

```
$SET FREEFORM SEQXEQ RESET LIST
FILE  1=LOCSAP/DATA,UNIT=DISK,SAVE=30,BLOCKING=3,RECORD=10
FILE  3=D/S,UNIT=REMOTE,LOCK,RECORD=9
FILE  4=LOCSAP/TIMEPLT,UNIT=DISK,SAVE=30,LOCK,AREA=200,
-     BLOCKING=3,RECORD=10
C-    ***********  SORSE FILE IS LOCSAP/JAN0275  **********
      REAL KT,KTORG,KTFIN,KZD,KZDORG,KZDFIN,KZ,KZORG,KZFIN
      REAL KTDORG,KTD,KTDFIN,NVAL
      DIMENSION RTDE1(7),RTE2(7),RZDE3(7),RZZO(8),OZDE3(6),OTDE1(6)
      DIMENSION OTE2(6)
      DIMENSION OTDE1R(6),OTDE1I(6),OTE2R(6),OTE2I(6),OZDE3R(6),
-     OZDE3I(6),TITLE(11)
      RADIAN = 180/(4*ATAN(1))
      READ(1,4000) (TITLE(I),I=1,11)
 4000 FORMAT(11A6)
      WRITE(3,4001) (TITLE(I),I=1,11)
 4001 FORMAT(//,11A6)
      WRITE(3,3101)
 3101 FORMAT(//,"PRINT STAB DERIV, DIMRTS, TFCPRT...")
      READ (3,/,END=99)NVAL,IDIMRTS,ITFCPRT; ISDPRNT=NVAL
      REAL    IY,LB,M,MU,MW,MTH,MQ,MUD,MWD,MQD,MX,MZ,MDELT,MTHUSQ
      READ(1,/,END=99) U,LB,ICABLE
  104 FORMAT(/////)
  301 READ(1,/,END=99)XU,ZU,MU,XW,ZW,MW,XTHUSQ,ZTHUSQ,MTHUSQ,XQ,ZQ,MQ,
-      XUD,ZUD, MUD,XWD,ZWD,MWD,XQD,ZQD,MQD,XX,ZX,MX,XZ,ZZ,MZ,XDELT,
-      ZDELT,MDELT,M,IY;XTH=XTHUSQ/(U*U);ZTH=ZTHUSQ/(U*U)
      MTH=MTHUSQ/(U*U)
  311 IF (ISDPRNT.EQ.0) GO TO 351
      WRITE (3,321)
  321 FORMAT(//20X,"S N A M E  NON-DIMENSIONAL")
      WRITE (3,331)
  331 FORMAT(16X,"LONGITUDINAL STABILITY DERIVATIVES"//)
      WRITE(3,341)XU,ZU,MU,XW,ZW,MW,XTH,ZTH,MTH,XQ,ZQ,MQ,XUD,ZUD,MUD,XWD
-     ,ZWD,MWD,XQD,ZQD,MQD,XX,ZX,MX,XZ,ZZ,MZ,XDELT,ZDELT,MDELT,M,IY,U,LB
  341 FORMAT(1X,"XU    =",E12.5,4X,"ZU    =",E12.5,4X,"MU    =",E12.5/,
-           1X,"XW    =",E12.5,4X,"ZW    =",E12.5,4X,"MW    =",E12.5/,
-           1X,"XTH   =",E12.5,4X,"ZTH   =",E12.5,4X,"MTH   =",E12.5/,
-           1X,"XQ    =",E12.5,4X,"ZQ    =",E12.5,4X,"MQ    =",E12.5/,
```

```
     -          1X,"XUD   =",E12.5,4X,"ZUD   =",E12.5,4X,"MUD   =",E12.5/,
     -          1X,"XWD   =",E12.5,4X,"ZWD   =",E12.5,4X,"MWD   =",E12.5/,
     -          1X,"XQD   =",E12.5,4X,"ZQD   =",E12.5,4X,"MQD   =",E12.5/,
     -          1X,"XX    =",E12.5,4X,"ZX    =",E12.5,4X,"MX    =",E12.5/,
     -          1X,"XZ    =",E12.5,4X,"ZZ    =",E12.5,4X,"MZ    =",E12.5/,
     -          1X,"XDELT =",E12.5,4X,"ZDELT =",E12.5,4X,"MDELT =",E12.5//
     -         ,9X,"M     =",E12.5,8X,"IY    =",E12.5//,
     -          9X,"UO    =",E12.5,8X,"LB    =",E12.5//)
  351 CONTINUE
  110 FORMAT(/,"J=",10X,"ECONV=",E12.5)
        DIMENSION DS(8),XS(8),ZS(8).TS(8),ROOTR(8),ROOTI(8),ECONV(8)
      CONTINUE
      UO=U
      N=6
      NP1=N+1
      ZQ=ZQ+M
      DCST=LB
      IF(ICABLE.EQ.0) DCST=UO
C-
C-    NON-DIMENSIONAL D(S) COEFFICIENTS FOR TOWED VEHICLES
C-
      DS(7)=(M-XUD)*(M-ZWD)*(IY-MQD)-XWD*ZQD*MUD
     -XQD*ZUD*MWD-(M-XUD)*ZQD*MWD
     -XWD*ZUD*(IY-MQD)-XQD*(M-ZWD)*MUD
      DS(6)=-XU*(M-ZWD)*(IY-MQD)-(M-XUD)*ZW*(IY-MQD)-(M-XUD)*(M-ZWD)*MQ
     -XW*ZQD*MUD-XWD*ZQ*MUD-XWD*ZQD*MU
     -XQ*ZUD*MWD-XQD*ZU*MWD-XQD*ZUD*MW
     +XU*ZQD*MWD-(M-XUD)*ZQ*MWD-(M-XUD)*ZQD*MW
     -XW*ZUD*(IY-MQD)-XWD*ZU*(IY-MQD)+XWD*ZUD*MQ
     -XQ*(M-ZWD)*MUD+XQD*ZW*MUD-XQD*(M-ZWD)*MU
      DS(5)=-XX*(M-ZWD)*(IY-MQD)+XU*ZW*(IY-MQD)-(M-XUD)*ZZ*(IY-MQD)
     +XU*MQ*(M-ZWD)+(M-XUD)*ZW*MQ-(M-XUD)*(M-ZWD)*MTH
     -XZ*ZQD*MUD-XW*ZQ*MUD-XWD*ZTH*MUD-XW*MU*ZQD-XWD*ZQ*MU-XWD*ZQD*MX
     -XTH*ZUD*MWD-XQ*ZU*MWD-XQD*ZX*MWD-XQ*ZUD*MW-XQD*ZU*MW-XQD*ZUD*MZ
     +XX*ZQD*MWD+XU*ZQ*MWD-(M-XUD)*ZTH*MWD+XU*ZQD*MW
     -(M-XUD)*ZQ*MW-(M-XUD)*ZQD*MZ
     -XZ*ZUD*(IY-MQD)-XW*ZU*(IY-MQD)-XWD*ZX*(IY-MQD)
     +XW*ZUD*MQ+XWD*ZU*MQ+XWD*ZUD*MTH
```

```
-      -XTH*(M-ZWD)*MUD+XQ*ZW*MUD+XQD*ZZ*MUD-XQ*(M-ZWD)*MU
-      +XQD*ZW*MU-XQD*(M-ZWD)*MX
       DS(4)=XX*ZW*(IY-MQD)+XU*ZZ*(IY-MQD)+XX*(M-ZWD)*MQ-XU*ZW*MQ
-      +(M-XUD)*ZZ*MQ+XU*(M-ZWD)*MTH+(M-XUD)*ZW*MTH
-      -XZ*ZQ*MUD-XW*ZTH*MUD-XZ*ZQD*MU-XW*ZQ*MU
-      -XWD*ZTH*MU-XW*ZQD*MX-XWD*ZQ*MX
-      -XTH*ZU*MWD-XQ*ZX*MWD-XTH*ZUD*MW-XQ*ZU*MW
-      -XQD*ZX*MW-XQ*ZUD*MZ-XQD*ZU*MZ
-      +XX*ZQ*MWD+XU*ZTH*MWD+XX*ZQD*MW+XU*ZQ*MW-(M-XUD)*ZTH*MW
-      +XU*ZQD*MZ-(M-XUD)*ZQ*MZ
-      -XZ*ZU*(IY-MQD)-XW*ZX*(IY-MQD)+XZ*ZUD*MQ+XW*ZU*MQ
-      +XWD*ZX*MQ+XW*ZUD*MTH+XWD*ZU*MTH
-      +XTH*ZW*MUD+XQ*ZZ*MUD-XTH*(M-ZWD)*MU+XQ*ZW*MU
-      +XQD*ZZ*MU-XQ*(M-ZWD)*MX+XQD*ZW*MX
       DS(3)=XX*ZZ*(IY-MQD)-XX*ZW*MQ-XU*ZZ*MQ+XX*(M-ZWD)*MTH
-      -XU*ZW*MTH+(M-XUD)*ZZ*MTH
-      -XZ*ZTH*MUD-XZ*ZQ*MU-XW*ZTH*MU-XZ*ZQD*MX
-      -XW*ZQ*MX-XWD*ZTH*MX
-      -XTH*ZX*MWD-XTH*ZU*MW-XQ*ZX*MW-XTH*ZUD*MZ
-      -XQ*ZU*MZ-XQD*ZX*MZ
-      +XX*ZTH*MWD+XX*ZQ*MW+XU*ZTH*MW+XX*ZQD*MZ
-      +XU*ZQ*MZ-(M-XUD)*ZTH*MZ
-      -XZ*ZX*(IY-MQD)+XZ*ZU*MQ+XW*ZX*MQ+XZ*ZUD*MTH
-      +XW*ZU*MTH+XWD*ZX*MTH
-      +XTH*ZZ*MUD+XTH*ZW*MU+XQ*ZZ*MU-XTH*(M-ZWD)*MX
-      +XQ*ZW*MX+XQD*ZZ*MX
       DS(2)=-XX*ZZ*MQ-XX*ZW*MTH-XU*ZZ*MTH
-      -XZ*ZTH*MU-XZ*ZQ*MX-XW*ZTH*MX
-      -XTH*ZX*MW-XTH*ZU*MZ-XQ*ZX*MZ
-      +XX*ZTH*MW+XX*ZQ*MZ+XU*ZTH*MZ
-      +XZ*ZX*MQ+XZ*ZU*MTH+XW*ZX*MTH
-      +XTH*ZZ*MU+XTH*ZW*MX+XQ*ZZ*MX
       DS(1)=-XX*ZZ*MTH-XZ*ZTH*MX-XTH*ZX*MZ
-      +XX*ZTH*MZ+XZ*ZX*MTH+XTH*ZZ*MX
       WRITE (3,3)
     3 FORMAT (//,20X,"**** DENOMINATOR DS(J) ****")
       DO 4 J=1,NP1
     4 IF (DS(J).NE.0.0) GO TO 14
```

```
      WRITE (3,5)
    5 FORMAT (/,16X,"*****  DS(J) COEFFICIENTS ALL ZERO  *****")
      GO TO 20
    8 FORMAT (//(12X,"J = ",I3,10X,"DS = ",E20.12))
   11 FORMAT (/,15X,"J = ",I3,10X,"ECONV = ",E15.8)
   13 FORMAT (//(1X,"J = ",I3,9X,"ROOTR = ",E12.5,9X,"ROOTI = ",E12.5))
C-
C-    DIMENSIONALIZE THE COEFFICIENTS OF THE DENOMINATOR TRANSFER
C-    FUNCTION
C-
   14 DO 15 I=1,NP1
   15 DS(I)=DS(I)*((LB/U)**(I-1))
      IF (ITFCPRT.NE.1) GO TO 17
      WRITE (3,16)
   16 FORMAT (//,22X,"DIMENSIONAL COEFFICIENTS")
      WRITE (3,8) ((J,DS(J)),J=1,NP1)
   17 CALL PRNBM (N,DS,ROOTR,ROOTI,ECONV)
      DO 18 J=1,N
   18 IF (ECONV(J).GT..5E-09) WRITE(3,11) J,ECONV(J)
      IF(IDIMRTS.EQ.0) GO TO 1
      WRITE (3,19)
   19 FORMAT (//,25X,"DIMENSIONAL ROOTS")
      WRITE (3,13) ((J,ROOTR(J),ROOTI(J)),J=1,N)
   20 WRITE (3,21)
   21 FORMAT (//)
    1 N=4
      NP1=N+1
C-
C-    NON-DIMENSIONAL X(S) COEFFICIENTS FOR TOWED VEHICLES
C-
      XS(5)=(M-ZWD)*(IY-MQD)*XDELT+XWD*ZQD*MDELT+XQD*MWD*ZDELT
     - -ZQD*MWD*XDELT+XWD*(IY-MQD)*ZDELT+XQD*(M-ZWD)*MDELT
      XS(4)=-ZW*(IY-MQD)*XDELT-(M-ZWD)*MQ*XDELT
     - +XW*ZQD*MDELT+XWD*ZQ*MDELT
     - +XQ*MWD*ZDELT+XQD*MW*ZDELT
     - -ZQ*MWD*XDELT-ZQD*MW*XDELT
     - +XW*(IY-MQD)*ZDELT-XWD*MQ*ZDELT
     - +XQ*(M-ZWD)*MDELT-XQD*ZW*MDELT
```

```
      XS(3)=-ZZ*(IY-MQD)*XDELT+ZW*MQ*XDELT-(M-ZWD)*MTH*XDELT
     -      +XZ*ZQD*MDELT+XW*ZQ*MDELT+XWD*ZTH*MDELT
     -      +XTH*MWD*ZDELT+XQ*MW*ZDELT+XQD*MZ*ZDELT
     -      -ZTH*MWD*XDELT-ZQ*MW*XDELT-ZQD*MZ*XDELT
     -      +XZ*(IY-MQD)*ZDELT-XW*MQ*ZDELT-XWD*MTH*ZDELT
     -      +XTH*(M-ZWD)*MDELT-XQ*ZW*MDELT-XQD*ZZ*MDELT
      XS(2)=ZZ*MQ*XDELT+ZW*MTH*XDELT+XZ*ZQ*MDELT+XW*ZTH*MDELT
     -      +XTH*MW*ZDELT+XQ*MZ*ZDELT-ZTH*MW*XDELT-ZQ*MZ*XDELT
     -      -XZ*MQ*ZDELT-XW*MTH*ZDELT-XTH*ZW*MDELT-XQ*ZZ*MDELT
      XS(1)=ZZ*MTH*XDELT+XZ*ZTH*MDELT+XTH*MZ*ZDELT
     -      -ZTH*MZ*XDELT-XZ*MTH*ZDELT-XTH*ZZ*MDELT
      WRITE (3,2122)
 2122 FORMAT (//,23X,"**** X NUMERATOR ****")
      DO 25 J=1,NP1
   25 IF (XS(J).NE.0.0) GO TO 31
      WRITE (3,26)
   26 FORMAT (/,13X,"*****  XS(J) COEFFICIENTS ALL ZERO  *****")
      GO TO 35
   28 FORMAT (//(12X,"J = ",I3,10X,"XS = ",E20.12))
C-
C-    DIMENSIONALIZE THE COEFFICIENTS OF THE X NUMERATOR TRANSFER
C-    FUNCTION
C-
   31 DO 32 I=1,NP1
   32 XS(I)=XS(I)*DCST*((LB/U)**(I-1))
      IF (ITFCPRT.NE.1) GO TO 33
      WRITE (3,16)
      WRITE (3,28) ((J,XS(J)),J=1,NP1)
   33 CALL PRNRM (N,XS,ROOTR,ROOTI,ECONV)
      DO 34 I=1,N
   34 IF (ECONV(J).GT..5E-09) WRITE(3,11) J,ECONV(J)
      IF(IDIMRTS.EQ.0) GO TO 36
      WRITE (3,19)
      WRITE (3,13) ((J,ROOTR(J),ROOTI(J)),J=1,N)
   35 WRITE (3,21)
   36 N=4
      NP1=N+1
C-
```

```
C-     NON-DIMENSIONAL Z(S) COEFFICIENTS FOR TOWED VEHICLES
C-
       ZS(5)=(M-XUD)*(IY-MQD)*ZDELT+ZQD*MUD*XDELT+XQD*ZUD*MDELT
     - +(M-XUD)*ZQD*MDELT+ZUD*(IY-MQD)*XDELT-XQD*MUD*ZDELT
       ZS(4)=-XU*(IY-MQD)*ZDELT-(M-XUD)*MQ*ZDELT
     - +ZQ*MUD*XDELT+ZQD*MU*XDELT
     - +XQ*ZUD*MDELT+XQD*ZU*MDELT-XU*ZQD*MDELT+(M-XUD)*ZQ*MDELT
     - +ZU*(IY-MQD)*XDELT-ZUD*MQ*XDELT
     - -XQ*MUD*ZDELT-XQD*MU*ZDELT
       ZS(3)=-XX*(IY-MQD)*ZDELT+XU*MQ*ZDELT-(M-XUD)*MTH*ZDELT
     - +ZTH*MUD*XDELT+ZQ*MU*XDELT+ZQD*MX*XDELT+XTH*ZUD*MDELT
     - +XQ*ZU*MDELT+XQD*ZX*MDELT
     - -XX*ZQD*MDELT-XU*ZQ*MDELT+(M-XUD)*ZTH*MDELT
     - +ZX*(IY-MQD)*XDELT-ZU*MQ*XDELT-ZUD*MTH*XDELT
     - -XTH*MUD*ZDELT-XQ*MU*ZDELT-XQD*MX*ZDELT
       ZS(2)=XX*MQ*ZDELT+XU*MTH*ZDELT+ZTH*MU*XDELT+ZQ*MX*XDELT
     - +XTH*ZU*MDELT+XQ*ZX*MDELT-XX*ZQ*MDELT-XU*ZTH*MDELT
     - -ZX*MQ*XDELT-ZU*MTH*XDELT-XTH*MU*ZDELT-XQ*MX*ZDELT
       ZS(1)=XX*MTH*ZDELT+ZTH*MX*XDELT+XTH*ZX*MDELT
     - -XX*ZTH*MDELT-ZX*MTH*XDELT-XTH*MX*ZDELT
       WRITE (3,350)
  350 FORMAT (//,23X,"***** Z NUMERATOR ****")
       DO 39 J=1,NP1
   39 IF (ZS(J).NE.0.0) GO TO 45
       WRITE (3,40)
   40 FORMAT (/,12X,"***** ZS(J) COEFFICIENTS ALL ZERO *****")
       GO TO 49
   42 FORMAT (//(12X,"J = ",I3,10X,"ZS = ",E20.12))
C-
C-     DIMENSIONALIZE THE COEFFICIENTS OF THE Z NUMERATOR TRANSFER
C-     FUNCTION
C-
   45 DO 46 I=1,NP1
   46 ZS(I)=ZS(I)*DCST*((LB/U)**(I-1))
       IF (ITFCPRT.NE.1) GO TO 47
       WRITE (3,16)
       WRITE (3,42) ((J,ZS(J)),J=1,NP1)
   47 CALL PRNBM (N,ZS,ROOTR,ROOTI,FCONV)
```

```
      DO 48 J=1,N
   48 IF (ECONV(J).GT..5E-09) WRITE(3,11) J,ECONV(J)
      IF(IDIMRTS.EQ.0) GO TO 50
      WRITE (3,19)
      WRITE (3,13) ((J,ROOTR(J),ROOTI(J)),J=1,N)
   49 WRITE (3,21)
   50 N=4
      NP1=N+1
C-
C-    NON-DIMENSIONAL T(S) COEFFICIENTS FOR TOWED VEHICLES
C-
      TS(5)=(M-XUD)*(M-ZWD)*MDELT+XWD*MUD*ZDELT+ZUD*MWD*XDELT
     - +(M-XUD)*MWD*ZDELT-XWD*ZUD*MDELT+(M-ZWD)*MUD*XDELT
      TS(4)=-XU*(M-ZWD)*MDELT-(M-XUD)*ZW*MDELT
     - +XW*MUD*ZDELT+XWD*MU*ZDELT+ZU*MWD*XDELT+ZUD*MW*XDELT
     - -XU*MWD*ZDELT+(M-XUD)*MW*ZDELT-XW*ZUD*MDELT-XWD*ZU*MDELT
     - -ZW*MUD*XDELT+(M-ZWD)*MU*XDELT
      TS(3)=-XX*(M-ZWD)*MDELT+XU*ZW*MDELT-(M-XUD)*ZZ*MDELT
     - +XZ*MUD*ZDELT+XW*MU*ZDELT+XWD*MX*ZDELT
     - +ZX*MWD*XDELT+ZU*MW*XDELT+ZUD*MZ*XDELT
     - -XX*MWD*ZDELT-XU*MW*ZDELT+(M-XUD)*MZ*ZDELT
     - -XZ*ZUD*MDELT-XW*ZU*MDELT-XWD*ZX*MDELT
     - -ZZ*MUD*XDELT-ZW*MU*XDELT+(M-ZWD)*MX*XDELT
      TS(2)=XX*ZW*MDELT+XU*ZZ*MDELT+XZ*MU*ZDELT+XW*MX*ZDELT
     - +ZX*MW*XDELT+ZU*MZ*XDELT-XX*MW*ZDELT-XU*MZ*ZDELT
     - -XZ*ZU*MDELT-XW*ZX*MDELT-ZZ*MU*XDELT-ZW*MX*XDELT
      TS(1)=XX*ZZ*MDELT+XZ*MX*ZDELT+ZX*MZ*XDELT
     - -XX*MZ*ZDELT-XZ*ZX*MDELT-ZZ*MX*XDELT
      WRITE (3,52)
   52 FORMAT (//,23X,"**** T NUMERATOR ****")
      DO 53 J=1,NP1
   53 IF (TS(J).NE.0.0) GO TO 59
      WRITE (3,54)
   54 FORMAT (/,13X,"*****  TS(J) COEFFICIENTS ALL ZERO  *****")
      GO TO 63
   56 FORMAT (//(12X,"J = ",I3,10X,"TS = ",E20.12))
C-
C-    DIMENSIONALIZE THE COEFFICIENTS OF THE T NUMERATOR TRANSFER
```

```
C-     FUNCTION
C-
   59 DO 60 I=1,NP1
   60 TS(I)=TS(I)*((LB/U)**(I-1))
      IF (ITFCPRT.NE.1) GO TO 61
      WRITE (3,16)
      WRITE (3,56) ((J,TS(J)),J=1,NP1)
   61 CALL PRNBM (N,TS,ROOTR,ROOTI,ECONV)
      DO 62 I=1,N
   62 IF (ECONV(J).GT..5E-09) WRITE(3,11) J,ECONV(J)
      IF(IDIMRTS.EQ.0) GO TO 63
      WRITE (3,19)
      WRITE (3,13)((J,ROOTR(J),ROOTI(J)),J=1,N)
   63 CONTINUE
C-
C-
C-
      OTDE1(6)=TS(5)
      OTDE1(5)=TS(4)
      OTDE1(4)=TS(3)
      OTDE1(3)=TS(2)
      OTDE1(2)=TS(1)
      OTDE1(1)=0.0
      N=5$N10=N
      CALL PRNBM(N,OTDE1,OTDE1R,OTDE1I,ECONV)
      WRITE(3,3201)(OTDE1R(J10),OTDE1I(J10),J10=1,N)
 3201 FORMAT(/,"ZEROS OF TD/E1",/,4(F9.4,F9.4))
      N=4$N12=N
      DO 2000 I1=1,5
      OTE2(I1)=TS(I1)
 2000 CONTINUE
      CALL PRNBM(N,OTE2,OTE2R,OTE2I,ECONV)
      WRITE(3,3202)(OTE2R(J12),OTE2I(J12),J12=1,N)
 3202 FORMAT(/,"ZEROS OF T/E2",/,4(F9.4,F9.4))
      OZDE3(6)=ZS(5)
      OZDE3(5)=ZS(4)-TS(5)*U0
      OZDE3(4)=ZS(3)-TS(4)*U0
      OZDE3(3)=ZS(2)-TS(3)*U0
```

```
      OZDE3(2)=ZS(1)-TS(2)*U0
      OZDE3(1)=-TS(1)*U0
      N=5$N14=N
      CALL PRNBM(N,OZDE3,OZDE3R,OZDE3I,ECONV)
      WRITE(3,3203)(OZDE3R(J14),OZDE3I(J14),J14=1,N)
3203  FORMAT(/,"ZEROS OF ZD/E3 AND Z/ZU",/,4(F9.4,F9.4))
2997  WRITE(3,2998)
2998  FORMAT("ENTER KTDORG,DELKTD,KTDFIN",/)
      READ(3,/)KTDORG,DELKTD,KTDFIN
      IF(DELKTD.EQ.0) GO TO 99
      NGAIN=(KTDFIN-KTDORG)/DELKTD+1
      KTD=KTDORG
      DO 3001 JD=1,NGAIN
      RTDE1(7)=DS(7)
      RTDE1(6)=DS(6)-KTD*TS(5)
      RTDE1(5)=DS(5)-KTD*TS(4)
      RTDE1(4)=DS(4)-KTD*TS(3)
      RTDE1(3)=DS(3)-KTD*TS(2)
      RTDE1(2)=DS(2)-KTD*TS(1)
      RTDE1(1)=DS(1)
      N=6
      CALL PRNBM(N,RTDE1,ROOTR,ROOTI,ECONV)
      DO 3006  J=1,N
      IF(ECONV(J) .GT. .5E-09) WRITE (3,110) J,ECONV(J)
3006  CONTINUE
      WRITE(3,3002)KTD
3002  FORMAT(F8.2)
      WRITE(3,3003)   (ROOTR(J),ROOTI(J),J=1,N)
3003  FORMAT(4(F9.4,F9.4))
3001  KTD=KTD+DELKTD
      WRITE(3,3004)
3004  FORMAT("DO YOU WANT TO CONTINUE KTD R. LOCUS")
      READ(3,/)NVAL
      IF (NVAL.EQ.1) GO TO 2997
3016  WRITE (3,3005)
3005  FORMAT("ENTER KTD")
      READ(3,/)KTD
3007   WRITE(3,3008)
```

```
 3008   FORMAT("ENTER KTORG,DELKT,KTFIN")
        READ(3,/)KTORG,DELKT,KTFIN
        IF(DELKT.EQ.0) GO TO 99
        NGAIN=(KTFIN-KTORG)/DELKT+1
        KT=KTORG
        DO 3012 JD=1,NGAIN
        RTE2(7)=DS(7)
        RTE2(6)=DS(6)-KTD*TS(5)
        RTE2(5)=DS(5)-KTD*TS(4)+KT*TS(5)
        RTE2(4)=DS(4)-KTD*TS(3)+KT*TS(4)
        RTE2(3)=DS(3)-KTD*TS(2)+KT*TS(3)
        RTE2(2)=DS(2)-KTD*TS(1)+KT*TS(2)
        RTE2(1)=DS(1)+KT*TS(1)
        N=6
        CALL PRNBM(N,RTE2,ROOTR,ROOTI.ECONV)
        DO 3009 J=1,N
        IF(ECONV(J) .GT. .5E-09) WRITE (3,110) J,ECONV(J)
 3009   CONTINUE
        WRITE(3,3010)KT
 3010    FORMAT(F8.2)
        WRITE(3,3011)((ROOTR(J),ROOTI(J),J=1,N))
 3011    FORMAT(4(F9.4,F9.4))
 3012    KT=KT+DELKT

        WRITE(3,3013)KTD
 3013   FORMAT("DO YOU WANT NEW KTD,0=NO,1=ENT,2=ENT&COMP,KTD NOW=",F8.2)
        READ(3,/)NVAL
        IF (NVAL.EQ.1)GO TO 3016; IF (NVAL.EQ.2) GO TO 2997
        WRITE(3,3014)
 3014   FORMAT("DO YOU WANT TO CONTINUE KT R. LOCUS")
        READ(3,/)NVAL
        IF (NVAL .EQ. 1) GO TO 3007
 3017   WRITE(3,3015)
 3015   FORMAT("ENTER KT")
        READ(3,/)KT
 3018    WRITE(3,3019)
 3019    FORMAT("ENTER KZDORG,DELKZD,KZDFIN")
        READ(3,/)KZDORG,DELKZD,KZDFIN
```

```
      IF(DELKZD.EQ.0) GO TO 99
      NGAIN=(KZDFIN-KZDORG)/DELKZD+1
      KZD=KZDORG
      DO 3022 JD=1,NGAIN
      RZDE3(7)=DS(7)
      RZDE3(6)=DS(6)-KTD*TS(5)+(KZD/RADIAN)*ZS(5)
      RZDE3(5)=DS(5)-KTD*TS(4)+KT*TS(5)+(KZD/RADIAN)*(ZS(4)-TS(5)*UU)
      RZDE3(4)=DS(4)-KTD*TS(3)+KT*TS(4)+(KZD/RADIAN)*(ZS(3)-TS(4)*UU)
      RZDE3(3)=DS(3)-KTD*TS(2)+KT*TS(3)+(KZD/RADIAN)*(ZS(2)-TS(3)*UU)
      RZDE3(2)=DS(2)-KTD*TS(1)+KT*TS(2)+(KZD/RADIAN)*(ZS(1)-TS(2)*UU)
      RZDE3(1)=DS(1)+KT*TS(1)+(KZD/RADIAN)*(-TS(1)*UU)
      N=6
      CALL PRNBM(N,RZDE3,ROOTR,ROOTI,ECONV)
      DO 3020 J=1,N
      IF (ECONV(J) .GT. .5E-09) WRITE (3,110) J,ECONV(J)
 3020  CONTINUE $WRITE(3,3110)KZD
      WRITE (3,3021)((ROOTR(J),ROOTI(J),J=1,N))
 3021  FORMAT (4(F9.4,F9.4))
 3022  KZD=KZD+DELKZD
      WRITE(3,3099)KTD
 3099 FORMAT("DO YOU WANT NEW KTD, KTD NOW =",F8.2)
      READ(3,/)NVAL
      IF (NVAL.EQ.1) GO TO 3016$ IF (NVAL.EQ.2) GO TO 2997
      WRITE(3,3023)KT
 3023 FORMAT("DO YOU WANT NEW KT, KT NOW =",F8.2)
      READ(3,/)NVAL
      IF (NVAL.EQ.1) GO TO 3017$IF (NVAL .EQ.2) GO TO 3007
      WRITE(3,3103)
 3103 FORMAT("DO YOU WANT TO CONTINUE KZD R. LOCUS")
      READ(3,/)NVAL
      IF (NVAL.EQ.1) GO TO 3018
 3024  WRITE(3,3025)
 3025 FORMAT("ENTER KZD")
      READ(3,/)KZD
 3026  WRITE(3,3098)
 3098  FORMAT("ENTER KZORG,DELKZ,KZFIN")
      READ(3,/)KZORG,DELKZ,KZFIN
      IF(DELKZ.EQ.0) GO TO 99
```

```
      NGAIN=(KZFIN-KZORG)/DELKZ+1
      KZ=KZORG
      DO 3029 JD=1,NGAIN
      RZZO(8)=DS(7)
      RZZO(7)=DS(6)-KTD*TS(5)+KZD*ZS(5)
      RZZO(6)=DS(5)-KTD*TS(4)+KT*TS(5)+KZD*(ZS(4)-TS(5)*UO)-KZ*ZS(5)
      RZZO(5)=DS(4)-KTD*TS(3)+KT*TS(4)+KZD*(ZS(3)-TS(4)*UO)-KZ*(ZS(4)
     -  -TS(5)*UO)
      RZZO(4)=DS(3)+KT*TS(3)+KZD*(ZS(2)-TS(3)*UO)-KZ*(ZS(3)-TS(4)*UO)
     -KTD*TS(2)
      RZZO(3)=DS(2)+KT*TS(2)+KZD*(ZS(1)-TS(2)*UO)-KZ*(ZS(2)-TS(3)*UO)
     -    -KTD*TS(1)
      RZZO(2)=DS(1)+KT*TS(1)-KZD*TS(1)*UO-KZ*(ZS(1)-TS(2)*UO)
      RZZO(1)=-KZ*(-TS(1)*UO)
       N=7
      CALL PRNBM(N,RZZO,ROOTR,ROOTI.ECONV)
      DO 3027 J=1,N
      IF (ECONV(J) .GT. .5E-09) WRITE (3,110) J,ECONV(J)
3027   CONTINUE $WRITE(3,3110)KZ
       GAIN=KZ*ZS(5)/DS(5)
        WRITE(3,3250)GAIN
3250   FORMAT("GAIN =",F8.4)
      WRITE (3,3028)((ROOTR(J),ROOTI(J),J=1,N))
3028   FORMAT (4(F9.4,F9.4))
3029   KZ=KZ+DELKZ
      WRITE(3,3099)KTD
      READ(3,/)NVAL
      IF (NVAL.EQ.1) GO TO 3016$IF (NVAL.EQ.2.) GO TO 2997
      WRITE(3,3023)KT
      READ(3,/)NVAL
      IF (NVAL.EQ.1) GO TO 3017$IF (NVAL .EQ.2) GO TO 3007
      WRITE(3,3032)KZD
3032  FORMAT("DO YOU WANT NEW KZD,KZD NOW =",F8.4)
      READ(3,/)NVAL
      IF (NVAL .EQ.1) GO TO 3024$ IF(NVAL.EQ.2)GO TO 3018
      WRITE(3,3033)
3033  FORMAT("DO YOU WANT TO CONTINUE KZ R. LOCUS")
      READ(3,/)NVAL
```

```
      IF (NVAL.EQ.1) GO TO 3026

      WRITE(3,9000)
 9000 FORMAT("DO YOU WANT A TIMEPLT")
      READ(3,/)NVAL
      IF(NVAL.EQ.0) GO TO 99

C-
      WRITE(4,4000) (TITLE(I),I=1,11)
C-
C-    STANDARD TEST VALUES USED FOR THESE TIMEPLTS
      IF=1
      TDEL=0.1
      TMAX=20.
      RTIME=0.
      PTIME=10.
      W=0
      IPLOT=1
      IWRITE=0

      WRITE(4,9003)IF,TDEL,TMAX,RTIME,PTIME,W,IPLOT,IWRITE
 9003 FORMAT(I1,1H,,F5.3,1H,,F8.3,1H,,F10.5,1H,,F10.5,1H,,
-     F10.5,1H,,I5,1H,,I5,2H,*)
   88 WRITE(3,9001)
 9001 FORMAT("ENTER KTD,KT,KZD,KZ,GAMP")
      READ(3,/)KTD,KT,KZD,KZ,GAMP
      WRITE(3,9002)
 9002 FORMAT("TIMEPLT FOR WHICH LOOP?",
-     ," ENTER 1=1ST, 2=2ND, 3=3RD, 4=4TH")
      READ(3,/)NVAL
      IF(NVAL.EQ.2) GO TO 9100
      IF(NVAL.EQ.3) GO TO 9101
      IF(NVAL.EQ.4) GO TO 9102
      GAIN=GAMP*(TS(5)/DS(7))
      WRITE(4,9004)N10,GAIN
 9004 FORMAT(I1,1H,,F15.5,2H,*)
      WRITE(4,9005)((DTDE1R(I1),DTDE1I(I1)),I1=1,N10)
 9005 FORMAT(4(F12.5,1H,),1H/)
```

```
      GO TO 9103
 9100 GAIN=GAMP*(TS(5)/DS(7))
      WRITE(4,9006)N12,GAIN
 9006 FORMAT(I1,1H,,F15.5,2H,*)
      WRITE(4,9007)((DTE2R(I2),DTE2I(I2)),I2=1,N12)
 9007 FORMAT(4(F12.5,1H,),1H/)
      GO TO 9103
 9101 GAIN=GAMP*(ZS(5)/DS(7))
      WRITE(4,9008)N14,GAIN
 9008 FORMAT(I1,1H,,F15.5,2H,*)
      WRITE(4,9009)((DZDE3R(I3),DZDE3I(I3)),I3=1,N14)
 9009 FORMAT(4(F12.5,1H,),1H/)
 9103 RZDE3(7)=DS(7)
      RZDE3(6)=DS(6)-KTD*TS(5)+KZD*ZS(5)
      RZDE3(5)=DS(5)-KTD*TS(4)+KT*TS(5)+KZD*(ZS(4)-TS(5)*U0)
      RZDE3(4)=DS(4)-KTD*TS(3)+KT*TS(4)+KZD*(ZS(3)-TS(4)*U0)
      RZDE3(3)=DS(3)+KT*TS(3)+KZD*(ZS(2)-TS(3)*U0)-KTD*TS(2)
      RZDE3(2)=DS(2)+KT*TS(2)+KZD*(ZS(1)-TS(2)*U0)-KTD*TS(1)
            RZDE3(1)=DS(1)+KT*TS(1)-KZD*TS(1)*U0
      N=6
      CALL PRNBM(N,RZDE3,ROOTR,ROOTI,ECONV)
      DO 9010 J=1,N
      IF (ECONV(J) .GT. .5E-09) WRITE (3,110) J,ECONV(J)
 9010 CONTINUE
      WRITE(4,9011)N
 9011 FORMAT(I1,2H,*)
      WRITE(4,9012)((ROOTR(J6),ROOTI(J6)),J6=1,N)
 9012 FORMAT(4(F12.5,1H,),1H/)
      GO TO 77

 9102 GAIN=GAMP*((-KZ*ZS(5))/DS(7))
      WRITE(4,9013)N14,GAIN
 9013 FORMAT(I1,1H,,F15.5,2H,*)
      WRITE(4,9014)((DZDE3R(J7),DZDE3I(J7)),J7=1,N14)
 9014 FORMAT(4(F12.4,1H,),1H/)
      RZZO(8)=DS(7)
      RZZO(7)=DS(6)-KTD*TS(5)+KZD*ZS(5)
      RZZO(6)=DS(5)-KTD*TS(4)+KT*TS(5)+KZD*(ZS(4)-TS(5)*U0)-KZ*ZS(5)
```

```
      RZZ0(5)=DS(4)-KTD*TS(3)+KT*TS(4)+KZD*(ZS(3)-TS(4)*UU)-KZ*(ZS(4)
     -  -TS(5)*UU)
      RZZ0(4)=DS(3)+KT*TS(3)+KZD*(ZS(2)-TS(3)*UU)-KZ*(ZS(3)-TS(4)*UU)
     - -KTD*TS(2)
      RZZ0(3)=DS(2)+KT*TS(2)+KZD*(ZS(1)-TS(2)*UU)-KZ*(ZS(2)-TS(3)*UU)
     -   -KTD*TS(1)
      RZZ0(2)=DS(1)+KT*TS(1)-KZD*TS(1)*UU-KZ*(ZS(1)-TS(2)*UU)
      RZZ0(1)=-KZ*(-TS(1)*UU)
       N=7
      CALL PRNBM(N,RZZ0,ROOTR,ROOTI,ECONV)
      DO 9015 J=1,N
      IF (ECONV(J) .GT. .5E-09) WRITE (3,110) J,ECONV(J)
 9015 CONTINUE
      WRITE(4,9016)N
 9016 FORMAT(I1,2H,*)
      WRITE(4,9017)((ROOTR(J9),ROOTI(J9)),J9=1,N)
 9017 FORMAT(4(F12.5,1H,),1H/)
   77 WRITE(3,9018)
 9018 FORMAT("DO YOU WANT ANOTHER TIMEPLT")
      READ(3,/)NVAL
      IF(NVAL.EQ.1) GO TO 88
   99 CONTINUE
C-
C-    NOW WRITE PROGRAM COST INFORMATION
C-
      IT1=TIME(2)/60.0; XM1=0.05*IT1
      IT2=TIME(3)/60.0; XM2=0.03*IT2
      IT3=TIME(7)/60.0; XM3=0.01*IT3
      XM4=XM1+XM2+XM3
      WRITE(3,51) IT1,XM1,IT2,XM2,IT3,XM3,XM4
   51 FORMAT (/," PROCESSOR TIME = ",I5," SEC",5X,"$",F6.2,/
     -      " I/O TIME       = ",I5," SEC",5X,"$",F6.2,/
     -      " PRORATED TIME  = ",I5," SEC",5X,"$",F6.2,/
     -      20X,"TOTAL COST  ","$",F6.2)
      STOP;END
      SUBROUTINE PRNBM (N,A,U,V,ECONV)
      DIMENSION A(8),U(8),V(8),ECONV(8),H(8),B(8),C(8)
      ICOUNT=1
```

```
      EFIX=.5E-09
      CONV=1.E-35
      NC=N+1
C-
C-    SEND COEFFICIENTS TO REDUCED COEFFICIENT STORAGE
C-
      DO 1 I=1,NC
      ECONV(I)=0.0
    1 H(I)=A(I)
C-
C-    INITIALIZE GUESSES AND SET REVERSAL INDICATOR NORMAL
C-
      P=0.
      Q=0.0
      R=0.
      IREV=1
C-
C-    SCALING TO BE DONE AT THIS POINT AND REMOVE ALL ZERO ROOTS
C-
    2 IF(H(NC)) 4,3,4
    3 NC=NC-1
      V(NC)=0.0
      U(NC)=0.0
      GO TO 2
    4 IF(H(1)) 7,5,7
    5 NC=NC-1
      V(NC)=0.
      U(NC)=0.
      DO 6 I=1,NC
    6 H(I)=H(I+1)
      GO TO 4
C-
C-    TEST FOR VARIOUS DEGREES
C-
    7 IF(ICOUNT.LT.2) GO TO 8
      ECONV(ICOUNT-1)=E
    8 ICOUNT=ICOUNT+1
    9 IF(NC-1) 10,50,10
```

```
   10 IF(NC-2) 12,11,12
   11 R=-H(1)/H(2)
      GO TO 37
   12 IF(NC-3) 14,13,14
   13 P=H(2)/H(3)
      Q=H(1)/H(3)
      GO TO 42
C-
C-    TEST TO REVERSE COEFFICIENTS AND DO SO IF TEST SUCCEEDS
C-
   14  IF(ABS(H(NC-1)/H(NC))-ABS(H(2)/H(1))) 15,21,21
   15 IREV=-IREV
      M=NC/2
      DO 16 I=1,M
      NL=NC+1-I
      F=H(NL)
      H(NL)=H(I)
   16 H(I)=F
      IF(Q) 18,17,18
   17 P=0.
      GO TO 19
   18 P=P/Q
      Q=1./Q
   19 IF(R) 20,21,20
   20 R=1./R
C-
C-    NEWTON, CALCULATE F(R) AND TEST FOR ROOT
C-
   21 E=EFIX
      B(NC)=H(NC)
      C(NC)=H(NC)
      B(NC+1)=0.
      C(NC+1)=0.
      NP=NC-1
   22 DO 35 J=1,1000
      DO 23 I1=1,NP
      I=NC-I1
      B(I)=H(I)+R*B(I+1)
```

```
   23 C(I)=B(I)+R*C(I+1)
      IF(ABS(B(1)/H(1))-E) 37,37,24
   24 IF(C(2)) 26,25,26
   25 R=R+1.
      GO TO 27
   26 R=R-B(1)/C(2)
C-
C-    MAKE A BAIRSTOW REDUCTION AND CORRECT
C-
   27 DO 28 I1=1,NP
      I=NC-I1
      B(I)=H(I)-P*B(I+1)-Q*B(I+2)
   28 C(I)=B(I)-P*C(I+1)-Q*C(I+2)
C-
C-    TEST FOR CONVERGENCE OF BAIRSTOW PROCESS
C-
      IF(H(2)) 30,29,30
   29 IF(ABS(B(2)/H(1))-E) 31,31,32
   30 IF(ABS(B(2)/H(2))-E) 31,31,32
   31 IF(ABS(B(1)/H(1))-E) 42,42,32
   32 CBAR=C(2)-B(2)
      D=C(3)**2-CBAR*C(4)
      IF(D) 34,33,34
   33 P=P-2.
      Q=Q*(Q+1.)
      GO TO 35
   34 P=P+(B(2)*C(3)-B(1)*C(4))/D
      Q=Q+(-B(2)*CBAR+B(1)*C(3))/D
   35 CONTINUE
      E=E*10.
      IF(E-CONV) 22,22,36
   36 CONV=E
      GO TO 42
C-
C-    LINEAR, COMPUTE AND STORE LINEAR ROOTS
C-
   37 NC=NC-1
      V(NC)=0.
```

```
      IF(IREV) 38,39,39
   38 U(NC)=1./R
      GO TO 40
   39 U(NC)=R
   40 DO 41 I=1,NC
   41 H(I)=H(I+1)
      GO TO 7
C-
C-    QUADRATIC.  SOLVE QUADRATIC AND STORE ROOTS
C-
   42 NC=NC-2
      IF(IREV) 43,44,44
   43 QP=1./Q
      PP=P/(Q*2.0)
      GO TO 45
   44 QP=Q
      PP=P/2.0
   45 F=(PP)**2-QP
      IF(F) 46,47,47
C-
C-    CASE OF IMAGINARY ROOTS.
C-
   46 U(NC+1)=-PP
      U(NC)=-PP
      V(NC+1)=SQRT(-F)
      V(NC)=-V(NC+1)
      GO TO 48
C-
C-    CASE OF REAL ROOTS
C-
   47 U(NC+1)=-SIGN(ABS(PP)+SQRT(F),PP)
      V(NC+1)=0.
      U(NC)=QP/U(NC+1)
      V(NC)=0.
C-
C-    FORM NEW REDUCED COEFFICIENTS
C-
   48 DO 49 I=1,NC
```

```
49 H(I)=B(I+2)
   GO TO 7
50 RETURN
   END
```

APPENDIX D

TIME HISTORY PLOT DIAGRAM CARD LIST

(Reverse Page D-2 Blank)

```
$CARD FREEFORM
FILE  1=TIMEPLT/DATA,UNIT=DISK,SAVE=30,LOCK,BLOCKING=3,RECORD=10
FILE  2=D/H,UNIT=REMOTE,LOCK,RECORD=9
FILE  3=PRIASP,UNIT=PRINTER,RECORD=15
FILE  4=FORP/DSK,UNIT=DISK,SAVE=30,LOCK,AREA=5000,RECORD=30
FILE  5=APER/DATA,UNIT=DISK,SAVE=30,LOCK,BLOCKING=3,RECORD=10
      REAL NG
      DIMENSION XLABEL(2),YLABEL(3),CONTRL(8),TITLE(11)
      DIMENSION NG(14),DG(14),ROOTR(14),ROOTI(14),OUTPUT(1002),T(1002),
-               FORCE(6,2),NC(5),NT(15),ZEROR(10),ZEROI(10)
      COMMON TDEL,TMAX
      DATA FORCE/"IMPU","STEP","RAMP","PULS","RAMP","SINU","LSE",
-                " "," ","E","STEP","SOID"/
      DATA(XLABEL(I),I=1,2)/12HTIME-(SEC)  /
      DATA(YLABEL(I),I=1,3)/18HOUTPUT-RESPONSE    /
C-   ****TIMEPLT/HUMP** SOURCE IS TIMEPLT/SORSE DR. D.F. HUMPHREYS
      ITF=0  ;  KK=-1
C-    99 FORMAT(//,"****** NOW,RUN THE PROGRAM PLOT/HUMP*****")
  100 FORMAT(8F10.3)
  105 FORMAT(4F20.4)
  106 FORMAT(5F15.3)
      READ(1,4000) (TITLE(I),I=1,11)
 4000 FORMAT(11A6)
      READ(1,/ )IF,TDEL,TMAX,RTIME,PTIME,W,IPLOT,IWRITE
    1 READ(1,/,END=3)INGM1,GAIN
      IF(INGM1.GT.0)GO TO 2
      ZEROR(1)=-1.  ;  GO TO 215
    2 READ(1,/)(ZEROR(I),ZEROI(I),I=1,INGM1)
  215 READ(1,/)IDGM1
      READ(1,/)(ROOTR(I),ROOTI(I),I=1,IDGM1)
      IF(INGM1.GT.1)GO TO 212
      NG(1)=-ZEROR(1)
      NG(2)=INGM1*1.
      ING=INGM1 + 1
      GO TO 4
  212 CALL CPMPY(ZEROR,ZEROI,INGM1,NG,ING)
    4 CONTINUE
C-
```

```
C-
C-
      KK=KK+1
      ITF=ITF + 1
  107 FORMAT(1H1)
      AMP=GAIN
C-
C-       IF - FORCING FUNCTION INDICATOR
C-                IF=0 IMPLIES AN IMPULSE
C-                IF=1 IMPLIES A STEP
C-                IF=2 IMPLIES A RAMP
C-                IF=3 IMPLIES A PULSE
C-                IF=4 IMPLIES A RAMPSTEP
C-                IF=5 IMPLIES A SINUSOID
      CALL        TIME(IF,AMP,PTIME,RTIME,W,OUTPUT,T,IO,NG,ING,ROOTR,
     -                 ROOTI,IDGM1,ICODE)
      WRITE(3,208)ITF
  208 FORMAT(/,1X, "*****TIME RESPONSE RUN NUMBER",I3," ******")
      WRITE(3,98)
   98 FORMAT(    /    ,10X,"THE COEFFICIENTS OF THE NUMERATOR"/)
      DO 70 I=1,ING
   70 WRITE(3,101)I,NG(I)
  101 FORMAT(10X,"NG(",I1,")=",F17.7)
      IF(INGM1.EQ.0)GO TO 216 $       WRITE(3,210)
  210 FORMAT(////10X,"THE ZEROS OF THE NUMERATOR"/)
      DO 82 I=1,INGM1
   82 WRITE(3,209)I,ZEROR(I),ZEROI(I)
  209 FORMAT(10X,"ZERO(",I1,")=",F17.7," + J ",F17.7)
  216 WRITE(3,102)
  102 FORMAT(////10X,"THE ROOTS OF THE DENOMINATOR"/)
      DO 80 I=1,IDGM1
   80 WRITE(3,103)I,ROOTR(I),ROOTI(I)
  103 FORMAT(10X,"ROOT(",I1,")=", F17.7," + J ",F17.7)
      WRITE(3,104) IF,(FORCE(IF+1,I),I=1,2),AMP
  104 FORMAT(////1X, "THE FORCING FUNCTION INDICATOR (IF) =",I3,/,
     -     " THIS IMPLIES THAT A ",2A4," INPUT WAS USED.",/,
     -                  " AMPLITUDE=",F17.7,//)
      IF(IF.EQ.3) WRITE(3,205) PTIME
```

```
  205 FORMAT( /20X,"PTIME=",F17.7)
      IF(IF.EQ.4) WRITE(3,204) RTIME
  204 FORMAT( /20X,"RTIME WAS ADJUSTED TO ",F10.4," SECONDS.")
      IF(IF.EQ.5) WRITE(3,206)W
  206 FORMAT(/ 20X,"FREQUENCY=",F17.7)
      IF (ICODE.EQ.0) GO TO 302;WRITE(3,96)ICODE
   96 FORMAT(//," ICODE =",I10)
      IF(ICODE.EQ.1) WRITE(3,201)
 201  FORMAT(//10X,"THE COMPLEX PART OF THE OUTPUT VECTOR BECAME",
-     "SIGNIFICANT")
      IF(ICODE.EQ.2)WRITE(3,202)
 202  FORMAT(//10X,"MULTIPLE ROOTS ENCOUNTERED")
      IF(ICODE.EQ.3) WRITE(3,203)
 203  FORMAT(//10X,"BAD ENTRY - CHECK POLYNOMIAL ORDERS OF T.F.")
      IF(ICODE.NE.0) GO TO 1
302   XMAX=T(1)
      XMIN=T(1)
      YMAX=OUTPUT(1)
      YMIN=OUTPUT(1)
      FLENUM=TMAX/TDEL
      IIH=FLENUM + 1.0
      DO 600 KI=1,IIH
      IF(T(KI).LT.XMIN)XMIN=T(KI)
      IF(T(KI).GT.XMAX)XMAX=T(KI)
      IF(OUTPUT(KI).GT.YMAX)YMAX=OUTPUT(KI)
      IF(OUTPUT(KI).LT.YMIN)YMIN=OUTPUT(KI)
  600 CONTINUE
      WRITE(3,97)
   97 FORMAT(1H1,////,8X,"TIME",13X,"OUTPUT",/)
      DO 10 I=1,IO,2
   10 WRITE(3,200)T(I),OUTPUT(I)
  200 FORMAT(2F17.7)
      WRITE(3,107)
      IF(IWRITE.EQ.0)GO TO 11
      ITER=1
      WRITE(5,213)
      WRITE(5,214)ITER,TMAX,IDGM1,IIH
  213 FORMAT(30X)
```

```
  214 FORMAT(I5,F10.1,2I5)
C-
   12 WRITE(5,207)(OUTPUT(I),I=1,ID)
  207 FORMAT(5(E12.5,1X))
   11 CONTINUE
      IF(IPLOT.EQ.0)GO TO 1
      DX=(XMAX-XMIN)/7. ; DY=(YMAX-YMIN)/5.
      IF(ABS(DY).GT..00001)GOTO334 ; WRITE(3,301) ; GOTO1
  334 IF(KK.GT.0)GO TO 700
      CALL PLOT(0,-12,-3)  ;  CALL PLOT(1,1,-3)  ;  GO TO 800
  700 CALL PLOT(12,0,-3)
  800 CONTINUE
  301 FORMAT(///,"  ******NO PLOT FOR LAST OUTPUT****** ")
      A=0. ; IF(YMIN.LT.0..AND.YMIN+DY*5.GT.0.)A=-YMIN/DY
      CALL AXIS(0,A,XLABEL,-11,7,0,XMIN,DX)
      CALL AXIS(0,0,YLABEL,15,5,90,YMIN,DY)
      CALL LINE(T,OUTPUT,IIH,1,0,3,XMIN,DX,YMIN,DY)
      GO TO 1
    3 CONTINUE  ;  IF(IPLOT.EQ.0)GO TO 217
      CALL PLOT(0,0,999)
            LOCK 4
C-          WRITE(2,107)  ;C-  WRITE(2,99)
  217 STOP
      END
      SUBROUTINE CPVAL(RES,ARG,X,IDIMX)
         COMPLEX RES,ARG
      DIMENSION X(20)
      RES=(0.,0.)
      J=IDIMX
    1 IF(J)3,3,2
    2 RES=RES*ARG+X(J)
      J=J-1
      GO TO 1
    3 RETURN
      END
C-*********************************************************************
C-
C-   PURPOSE"
```

```
C-
C-       THE PURPOSE OF THIS SUBROUTINE IS TO DETERMINE THE TIME
C-       RESPONSE OF AN INPUT TO A TRANSFER FUNCTION BY TAKING THE
C-       INVERSE LAPLACE TRANSFORM BY THE METHOD OF RESIDUES.
C-       INTRODUCTION TO AUTOMATIC CONTROL SYSTEMS - ROBERT N. CLARK
C-       (PP 70 - 77).
C-
C-
C-    VARIABLES"
C-       AMP - AMPLITUDE OF THE FORCING FUNCTION
C-       PTIME - PULSE TIME (SPECIFIED FOR IF=3)
C-       RTIME - RAMP TIME (SPECIFIED FOR IF=4)
C-       W - FREQUENCY OF THE SINUSOIDAL INPUT (SPECIFIED FOR IF=5)
C-       OUTPUT - VECTOR OF CALCULATED RESPONCE AMPLITUDE VALUES
C-       T - VECTOR OF SEQUENTIAL TIME VALUES DIRECTLY RELATED TO OUTPUT
C-       IO - NUMBER OF OUTPUT VALUES (CALCULATED)
C-       NG - VECTOR OF NUMERATOR COEFFICIENTS
C-       ING - DIMENSION OF THE NUMERATOR COEFFICIENTS
C-       ROOTR - VECTOR OF REAL PARTS OF THE ROOTS
C-       ROOTI - VECTOR OF IMAGINARY PARTS OF THE ROOTS
C-       IDGM1 - NUMBER OF ROOTS (ORDER OF THE DENOMINATOR)
C-       ICODE - RETURN CODE VARIABLE
C-               ICODE=0 IMPLIES NORMAL EXECUTION
C-               ICODE=1 IMPLIES THAT THE COMPLEX PART OF THE OUTPUT
C-                       VECTOR BECAME SIGNIFICANT.
C-
C-
C-               ICODE=3 IMPLIES THAT THE ORDER OF THE DENOMINATOR WAS
C-                       NOT GREATER THAN THE ORDER OF THE NUMERATOR.
C-               ICODE=4 IMPLIES THAT THE FORCING FUNCTION INDICATOR
C-                       WAS SPECIFIED INCORRECTLY
C-
C-
C-    SUBROUTINES CALLED"
C-
C-       CPVAL - COMPLEX EVALUATION OF A POLYNOMIAL
C-       TYME - RESPONSE BY THE METHOD OF RESIDUES
C-
```

```
C-
C-   REMARKS"
C-
C-       THIS SUBROUTINE IS DESIGNED TO GENERATE THE TIME RESPONSE OF A
C-       GENERAL OUTPUT FUNCTION XO(S) = X1(S)G(S).  IN THIS
C-       EVALUATION TWO IMPORTANT ASSUMPTIONS ARE MADE.
C-               1)  THE ORDER OF THE DENOMINATOR OF THE OUTPUT FUNCTION
C-                   OF THE OUTPUT FUNCTION.
C-                   MUST BE LARGER THAN THE ORDER OF THE NUMERATOR
C-               2)  MULTIPLE ROOTS OF THE DENOMINATOR MAY NOT EXIST.
C-
C-**********************************************************************
C-
      SUBROUTINE TIME(IF,AMP,PTIME,RTIME,W,OUTPUT,T,IO,NG,ING,ROOTR,
     -                ROOTI,IDGM1,ICODE)
         REAL NG
      DIMENSION P(14),NG(14),K(14),OUT(1002),T(1002),OUTPUT(1002),
     -          ROOTR(14),ROOTI(14),SAVE(1002)
      COMMON TDEL,TMAX
      ICODE=4
      IF(IF.GT.5.OR.IF.LT.0)RETURN
      GAINDG=1./AMP
C-
C-       DETERMINE TMAX
C-
C-      SMALL=1.E6
C-      DO  9 I=1,IDGM1
C-      ABSR=ABS(ROOTR(I))
C-      IF (ABSR.EQ.0.) GO TO 10
C-      IF(ABSR.LT.SMALL) SMALL=ABSR
C-
      IF(IF.EQ.0)GO TO 11
      GO TO (10,20,30,40,50) ,IF
   11 CALL TYME(OUTPUT,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)
      RETURN
   10 IDGM1=IDGM1+1
      ROOTR(IDGM1)=0.
      ROOTI(IDGM1)=0.
```

```
      CALL TYME(OUTPUT,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)
      RETURN
   20 IDGM1=IDGM1+1
      ROOTR(IDGM1)=.001
      ROOTI(IDGM1)=0.
      IDGM1=IDGM1+1
      ROOTR(IDGM1)=-.001
      ROOTI(IDGM1)=0.
      CALL TYME(OUTPUT,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)
      RETURN
   30 IDGM1=IDGM1+1
      ROOTR(IDGM1)=0.0
      ROOTI(IDGM1)=0.0
      CALL TYME(SAVE,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)
      IF(ICODE.NE.0) GO TO 34
      MTIME=PTIME/TDEL+0.5
      IF(MTIME.EQ.0) GO TO 32
      DO 31 I=1,MTIME
   31 OUTPUT(I)=SAVE(I)
   32 IP1=MTIME+1
      DO 33 I=IP1,IO
   33 OUTPUT(I)=SAVE(I)-SAVE(I-MTIME)
   34 RETURN
   40 I=RTIME/TDEL+.5
      RTIME=I*TDEL
      IDGM1=IDGM1+1
      ROOTR(IDGM1)=.001
      ROOTI(IDGM1)=0.0
      IDGM1=IDGM1+1
      ROOTR(IDGM1)=-.001
      ROOTI(IDGM1)=0.0
      GAINDG=GAINDG*RTIME
      CALL TYME(SAVE,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)
      IF(ICODE.NE.0) GO TO 44
      MTIME=RTIME/TDEL+0.5
      IF(MTIME.EQ.0) GO TO 42
      DO 41 I=1,MTIME
   41 OUTPUT(I)=SAVE(I)
```

```
42 IP1=MTIME+1
   DO 43 I=IP1,IO
43 OUTPUT(I)=SAVE(I)-SAVE(I-MTIME)
44 RETURN
50 IDGM1=IDGM1+1
   ROOTR(IDGM1)=0.
   ROOTI(IDGM1)=W
   IDGM1=IDGM1+1
   ROOTR(IDGM1)=0.
   ROOTI(IDGM1)=-W
   CALL TYME(OUTPUT,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)
   RETURN
C- DEBUG SUBCHK
   END
   SUBROUTINE TYME(OUTPUT,T,IO,NG,ING,ROOTR,ROOTI,IDGM1,GAINDG,ICODE)
     COMPLEX P,S,OUT,OUT1,K,KJ,KNUM
     REAL NG
   DIMENSION P(14),NG(14),K(14),OUT(1002),OUTPUT(1002),T(1002),
  -          ROOTR(14),ROOTI(14),TTEST(16)
   COMMON TDEL,TMAX
C-
C-    CHECK FOR BAD ENTRY
C-
   IF(IDGM1.LT.ING) GO TO 55
C-
C-    CHECK FOR MULTIPLE ROOTS
C-
   DO 5 I=1,IDGM1
   RRP1=ROOTR(I)+.0001
   RRM1=ROOTR(I)-.0001
   RIP1=ROOTI(I)+.0001
   RIM1=ROOTI(I)-.0001
   DO 5 J=1,IDGM1
   IF(I.EQ.J) GO TO 5
   RRJ=ROOTR(J)
   RIJ=ROOTI(J)
   IF(RRM1.LT.RRJ.AND.RRP1.GT.RRJ.AND.RIM1.LT.RIJ.AND.RIP1.GT.RIJ)
  -GO TO 50
```

```
      GO TO 5
   50 ROOTR(J)=ROOTR(J) - 0.0002
 5    CONTINUE
      ICODE=0
      DO 11 I=1,IDGM1
   11 P(I)=CMPLX(ROOTR(I),ROOTI(I))
C-
C-        DETERMINE THE K"S
C-
      DO 15 J=1,IDGM1
      S=P(J)
      CALL CPVAL(KNUM,S,NG,ING)
      KJ=(1.,0.)
      DO 25 L=1,IDGM1
      IF(L.EQ.J) GO TO 25
      KJ=KJ/(S-P(L))
   25 CONTINUE
      K(J)=KJ*KNUM/GAINDG
   15 CONTINUE
C-
C-        DETERMINE THE TIME RESPONSE
C-
      IO=0
      T1=-TDEL
   34 IO=IO+1
      OUT1=(0.,0.)
      T1=T1+TDEL
      DO 35 J=1,IDGM1
      IF(ROOTR(J)*T1.LT.-100.) GO TO 35
      OUT1=OUT1+K(J)*CEXP(T1*P(J))
   35 CONTINUE
      OUTPUT(IO)=REAL(OUT1)
      UNREAL=AIMAG(OUT1)
      IF(ABS(UNREAL).LT..01)GO TO 100
      WRITE(3,101)OUT1
  101 FORMAT(2F20.7)
  100 CONTINUE
      T(IO)=T1
```

```
      IF(T1.LT.TMAX) GO TO 34
      IF(ABS(UNREAL).GT..01)WRITE(3,102)
  102 FORMAT(1H1)
      RETURN
 55   ICODE=3
      RETURN
C=    DEBUG SUBCHK
      END
      SUBROUTINE PLOT(X,Y,IPEN)
      DIMENSION CONT(9)
      DATA CONT/51HCCFX PLOT/HUMP*577351002$FILE FILERAY=FORP/DSK$END./
C=
      BCD =(6H     P)
      WRITE(4)BCD,X,Y,IPEN$IF(IPEN.NE.999)RETURN$LOCK 4
      CALL ZIP(CONT)$RETURN
      END
      SUBROUTINE SYMBOL(X,Y,SZ,BCD,ANG,NC)
      DIMENSION BCD(13)
      T=(6H     S)
                   NW1=6
      IF(MOD(NC,6).EQ.0)NW1=0
      NW =(IABS(NC)+ NW1)/6
      WRITE(4)T,X,Y,SZ,ANG,NC,(BCD(I),I=1,NW)
      RETURN
      END
      SUBROUTINE AXIS(X,Y,BCD,NC,AXLEN,ANG,RMIN,DELTV)
      DIMENSION BCD(13)
      T =(6H     A)
                   NW1=6
      IF(MOD(NC,6).EQ.0)NW1=0
      NW =(IABS(NC) + NW1)/6
      WRITE(4)T,X,Y,NC,AXLEN,ANG,RMIN,DELTV,(BCD(I),I=1,NW)
      RETURN
      END
      SUBROUTINE LINE(PX,PY,NPT,INC,LTYP,ISM,FIRSTX,DELTX,FIRSTY,DELTY)
      DIMENSION PX(NPT),PY(NPT),BCD(1)
      INTEGER A,TEMP,B,C
      S =(6H     S)
```

```
      P =(6H      P)
      SZ =.08
      ANG =0.0 ; IX =1
   20 A=1
      B =IABS(INC)
       C=NPT
      NA = 0
      IC = 3
      IS =-1
      ICA = 2
      ISA =-2
      NT = 1
      IF(LTYP)30,40,50
   30 ISA = -1
      B =B *LTYP +ISA
      GO TO 60
   40 NT = -1
      GO TO 60
   50 NT =LTYP;NA=-1+NT
   60 IF(ISM=3)BCD(1)=(6H+00000)
      IF(ISM=1)BCD(1)=(6H000000)
      DO 100 I=A,C,B
      XPT =(PX(I) -FIRSTX)/DELTX
      YPT =(PY(I) -FIRSTY)/DELTY
      NA =NA+1
      IF(NA=NT)GO TO 110
      WRITE(4)P,XPT,YPT,IC
      GO TO 105
  110 WRITE(4)S,XPT,YPT,SZ,ANG,IX,BCD(1)
      NA =0
  105 IC =ICA
      IS =ISA
  100 CONTINUE
      RETURN
      END

 SUBROUTINE CPMPY(ZEROR,ZEROI,INGM1,A,IDIMZ)
 COMPLEX X,Y,Z
```

```
      DIMENSION X(28),Y(28),Z(28),ZEROR(28),ZEROI(28),A(28)
      DO 11 I=1,INGM1
11    ZEROR(I)=-ZEROR(I)
      X(1)=CMPLX(ZEROR(1),ZEROI(1))
      X(2)=(1.,0.)
      IDIMX=2
      Y(2)=(1.,0.)
      IDIMY=2
      MAXDO=INGM1 - 1
      DO 1 IO=1,MAXDO
      J=IO + 1
      Y(1)=CMPLX(ZEROR(J),ZEROI(J))
      IDIMZ=IDIMX + IDIMY - 1
      DO 30 I1=1,IDIMZ
30    Z(I1)=(0.,0.)
      DO 40 I2=1,IDIMX
      DO 40 J1=1,IDIMZ
      K=I2 + J1 - 1
40    Z(K)=X(I2)*Y(J1) + Z(K)
      DO 50 I3=1,IDIMZ
50    X(I3)=Z(I3)
1     IDIMX=IDIMZ
      DO 60 IA=1,IDIMZ
60    A(IA)=REAL(Z(IA))
      DO 20 J=1,INGM1
20    ZEROR(J)=-ZEROR(J)
      RETURN
      END
```

INITIAL DISTRIBUTION LIST
NCSL REPORT 242-75

	Commander, Naval Coastal Systems Laboratory	
	(Code 112.2 - Technical Library)	(Copies 1-4)
	(Code 741 - D. W. Shepherd)	(Copy 5)
	(Code 731 - R. E. Kelly)	(Copy 6)
	(Code 731 - R. Mallory)	(Copy 7)
	(Code 731 - C. A. Dildy)	(Copy 8)
	(Code 731 - A. M. Thomas)	(Copy 9)
	(Code 722 - A. G. Holston)	(Copy 10)
	(Code 796 - K. W. Watkinson)	(Copy 11)
	(Code 732 - T. C. Watson)	(Copy 12)
	(Code 796 - D. E. Humphreys)	(Copies 13-32)
	(Code 796 - R. W. Miller)	(Copies 33-37)
	(Code 710.2 - L. F. Dewberry)	(Copies 38-42)
	(Code 724 - A. S. Piper)	(Copy 43)
427	Commander, Naval Sea Systems Command	
	(SEA 03C)	(Copy 44)
	(SEA 0322 - Benen, Sejd)	(Copies 45-46)
	(SEA 0342 - Romano)	(Copy 47)
	(SEA 03512 - Peirce)	(Copy 48)
	(SEA 03513 - Pasiuk)	(Copy 49)
	(SEA 0353 - Freund)	(Copy 50)
	(SEA 036 - Smith)	(Copy 51)
	(SEA 06H1)	(Copy 52)
	(SEA 09G32)	(Copy 53)
	(SEA 660E)	(Copy 54)
	(PMS 395)	(Copy 55)
210	Commander, Naval Surface Weapons Center, White Oak	
	(Technical Library)	(Copy 56)
223	Director, Naval Research Laboratory	
	(Ocean Engineering Dept. - R. Skop)	(Copy 57)
	(Ocean Engineering Dept. - H. Johnson)	(Copy 58)
	(Technical Library)	(Copy 59)
186	Officer-in-Charge, Civil Engineering Laboratory	
	(Code L31)	(Copy 60)
198	Commander, Naval Facilities Engineering Command	
	(Code 032C)	(Copy 61)
266	Commanding Officer, Naval Underwater Systems Center	
	(Technical Library)	(Copy 62)
265	Commander, Naval Undersea Center	
	(Technical Library)	(Copy 63)
	(Code 65 - I. Lemaire)	(Copy 64)
	(Code 6512 - R. A. Marrone)	(Copy 65)
268	Commander, Naval Weapons Center	
	(Technical Library)	(Copy 66)
54	Chief of Naval Research	
	(Code 438 - R. D. Cooper)	(Copy 67)
-	Director, Office of Naval Research, Branch Office, 495 Summer St., Boston, MA 02210	
	(Dr. Al Wood)	(Copy 68)

INITIAL DISTRIBUTION LIST (Cont'd.)
NCSL REPORT 242-75

1	Chief of Naval Material	
	(MAT 03P)	(Copy 69)
	(MAT 031)	(Copy 70)
	(MAT 032)	(Copy 71)
3	Chief of Naval Operations	
	(OP 098 - RDT&E - VADM Weymouth)	(Copy 72)
	(OP 02 - Submarine Warfare, VADM Beshamy)	(Copy 73)
	(OP 095 - Dir., ASW Programs, VADM Shear)	(Copy 74)
	(OP 21 - Strategic Sub. Div. & ULMS Prog. Coord., RADM Kaufman)	(Copy 75)
	(OP 22 - Attach Sub. Div. & SSN Prog. Coord.)	(Copy 76)
78	Director of Defense Research & Engineering	
	(Research & Technology)	(Copy 77)
28	Assistant Secretary of the Navy	
	(Research & Development)	(Copy 78)
208	Commander, Naval Oceanographic Office	
	(Code 1640 - Library)	(Copy 79)
154	Superintendent, Naval Academy	
	(Technical Library)	(Copy 80)
222	Superintendent, Naval Postgraduate School	
	(Code 2124 - Library)	(Copy 81)
236	Commander, Naval Ship Research & Development Center	
	(Code 15)	(Copy 82)
	(Code 1564)	(Copy 83)
	(Code 1576)	(Copy 84)
75	Director, Defense Documentation Center	(Copies 85-96)
-	Library of Congress, Science & Technology Division, Washington, DC 20540	(Copy 97)
-	Department of Ocean Engineering, Mass. Institute of Technology, Cambridge, MA 02138	
	(Department Library)	(Copy 98)
	(Prof. M. Abkowitz)	(Copy 99)
-	Davidson Laboratory, Stevens Institute of Technology, 711 Hudson St., Hoboken, NJ 07030	(Copy 100)
-	Iowa Institute of Hydraulic Research, Univ. of Iowa, Iowa City, IA 52240	
	(Prof. L. Landweber)	(Copy 101)
-	(Prof. V. C. Patel)	(Copy 102)
-	Penn. State University, Ordnance Research Laboratory, P. O. Box 30, State College, PA 16801	
	(Library)	(Copy 103)
340	Director, Woods Hole Oceanographic Institute	
	(Ocean Engineering Department)	(Copy 104)
-	Dept. of Mechanical Engineering, Catholic University, Washington, DC	
	(Prof. M. J. Casarella)	(Copy 105)

- Oceanics, Inc., Technical Industrial Park, Plainview L.I., New York 11803 (Copy 106)
- Society of Naval Architects & Marine Engineers, 74 Trinity Place, New York 10006
  - (Technical Library) (Copy 107)
- Hydronautics, Inc., Pindell School Road, Howard County, Laurel, MD 20810
  - (Dr. B. L. Silverstein) (Copy 108)
  - (M. P. Tulin) (Copy 109)